Eksploracja logów procesów

Proccess mining

Process mining

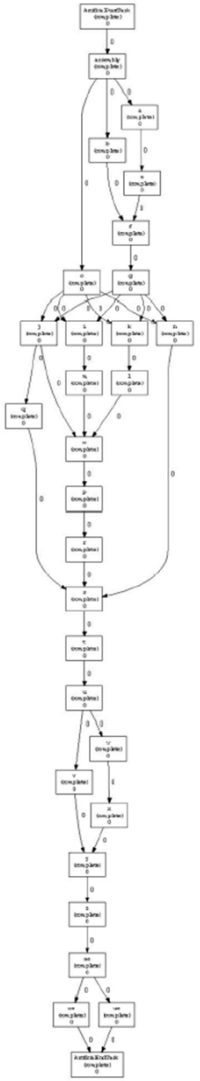
Celem eksploracji logów procesów biznesowych jest:

- Odkrywanie modelu procesów biznesowych
- Analiza procesów biznesowych
- Ulepszanie procesów biznesowych

Złożoność rzeczywistych procesów

Modele procesów,
a rzeczywiste wystąpienia procesów

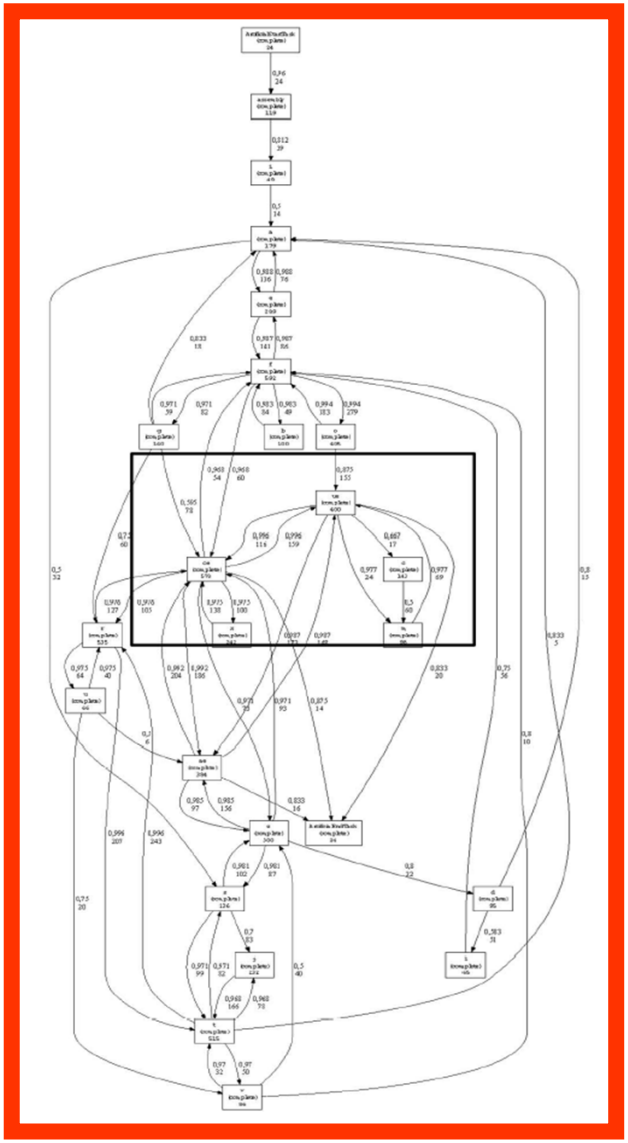
model



rzeczywistość

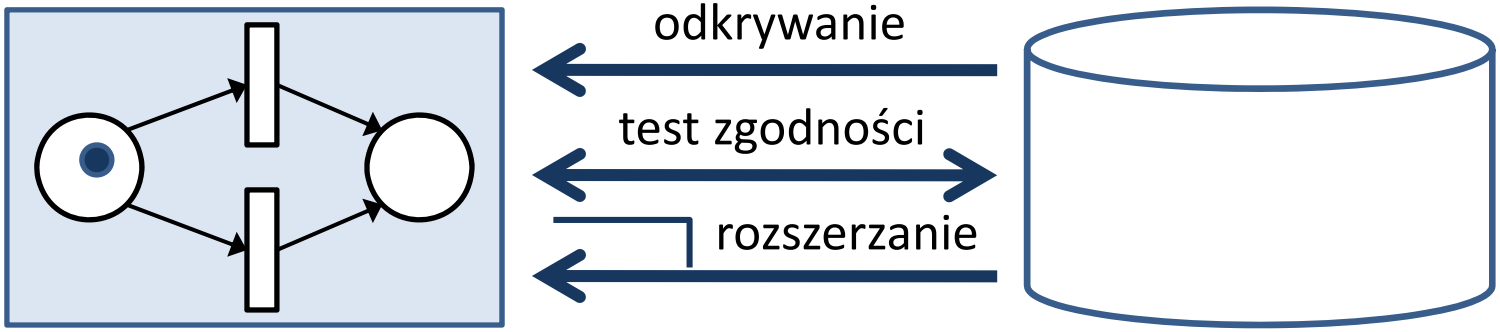
Wystąpienia zgodne z modelem: 37,5%

Wystąpienia niezgodne z modelem: 62,5%



Typy eksploracji procesów

- **Odkrywanie** – dla procesów, które nie posiadają zdefiniowanych modeli, model może być zdefiniowany jako wynik eksploracji zebranych logów; np. ProM tworzy automatycznie model sieci Petriego
- **Test zgodności** – modelu teoretycznego z rzeczywistym przebiegiem procesów wynikającym z logów procesów
- **Rozszerzenie** modelu teoretycznego o wiedzę wynikającą z logów procesów



Informacje przechowywane w logach

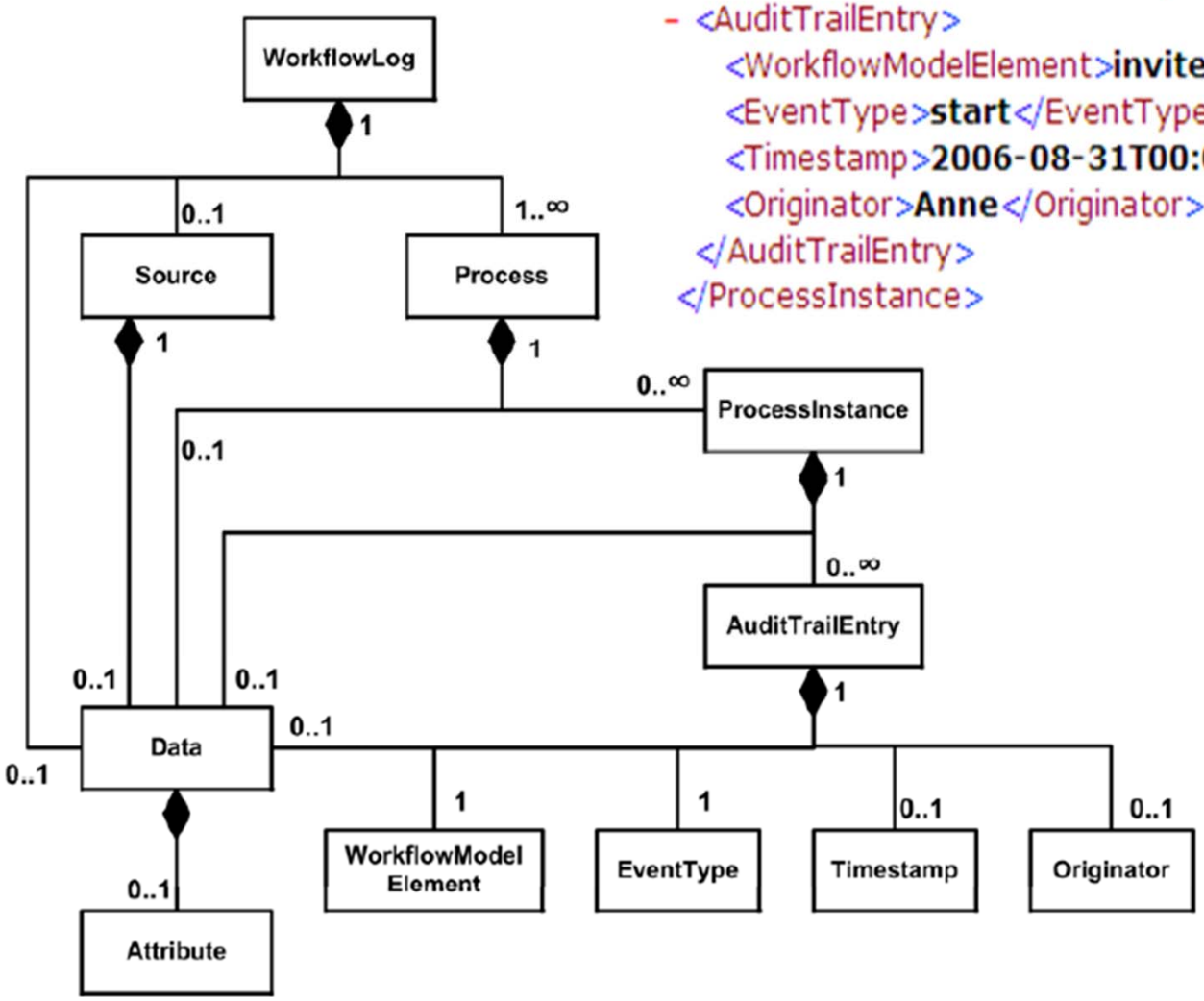
Trzy podstawowe perspektywy analizy danych

- Organizacyjna - kto uczestniczył w realizacji procesu
- Procesowa – jak wykonywane są procesy
- Przypadków – co dzieje się w firmie

Co? Jak? Kto?

Id wystąpienia	Id zadania	Wykonawca	Czas
107	A	Nowak	7-1-2008 9:45
108	B	Kowalski	7-1-2008 10:05
109	A	Tarzan	7-1-2008 12:17
110	C	Buła	7-1-2008 14:48
107	B	Kowalski	8-1-2008 8:15
110	E	Nowak	8-1-2008 9:32
108	C	Buła	8-1-2008 11:15

Zawartość logu ProM

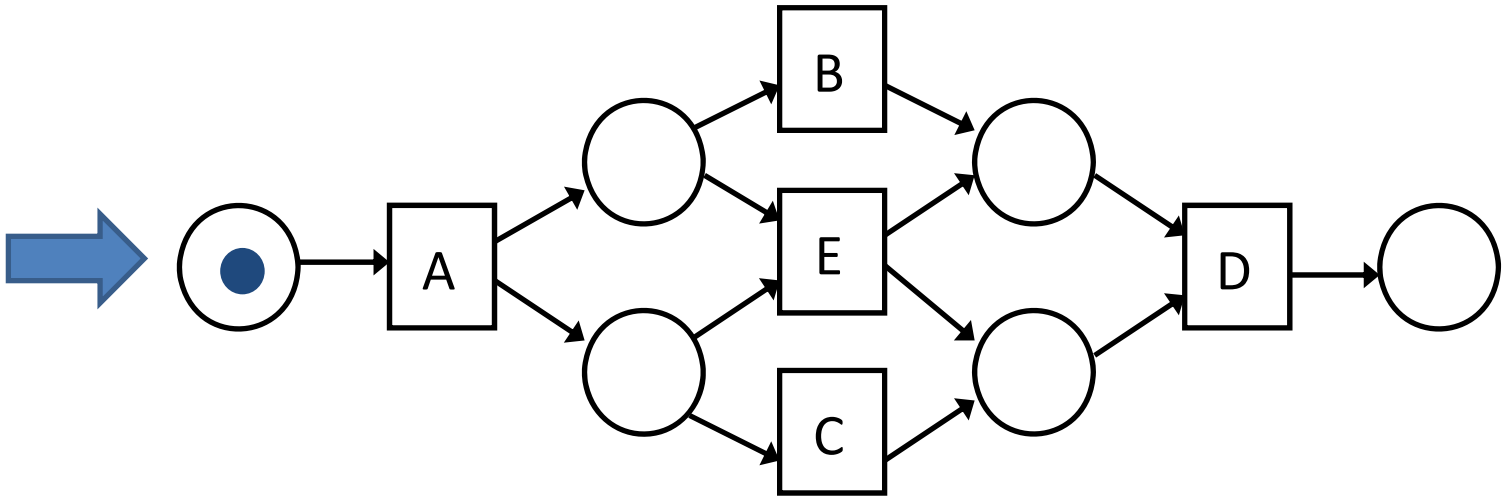



```
<ProcessInstance id="52" description="">
- <AuditTrailEntry>
    <WorkflowModelElement>invite reviewers</WorkflowModelElement>
    <EventType>start</EventType>
    <Timestamp>2006-08-31T00:00:00.000+01:00</Timestamp>
    <Originator>Anne</Originator>
  </AuditTrailEntry>
 </ProcessInstance>
```

Odkrywanie modelu procesów

Na podstawie występujących sekwencji zadań jest odtwarzany model przepływu pracy

ABCD
ACBD
AED
ABCD
ABCD
AED
ACBD



Własności procesu odkrywania

- **Kompletność**: Uzyskany model powinien zachować wszystkie zależności między zadaniami, które da się wywieźć z logu. Powinna również umożliwiać wykonanie wszystkie wystąpień procesów zapamiętanych w logu.
- **Poprawność**: Model nie powinien wprowadzać fałszywych zależności między zadaniami.
- **Minimalność**: Dla większej czytelności, model powinien mieć minimalną liczbę przepływów.

Algorytm α

Relacje występujące między zadaniami

- **Bezpośrednie następstwo**
  - x > y jeżeli w logu istnieje wystąpienie procesu, w którym zadnie x występuje bezpośrednio przed zadaniem y.
- **Przyczynowość**
  - x → y jeżeli istnieją wystąpienia procesów, w których występuje x > y, a nie ma wystąpień, w których występuje y > x.
- **Współbieżność**
  - x||y jeżeli istnieją wystąpienia procesów, w których występuje x > y i równocześnie istnieją wystąpienia procesów, w których występuje y > x
- **Wybór**
  - x#y jeżeli w żadnym wystąpieniu nie ma przypadków x>y, ani y>x.

Wystąpienie	Zadanie
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task E
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task F
case 4	task D

Algorytm α

Odkrywanie sekwencji w logu

- Sekwencje są projekcjami zadań należących do tego samego wystąpienia

case 1(Log) = ABCD
case 2(Log) = ABCD
…

ABCD - cases: 1, 2, 3; razem (3)
ACBD – case: 4; razem (1)
EF – case: 5; razem (1)

Algorytm α - relacje między zadaniami

Odkrywanie relacji między zadaniami

- Bezpośrednie następstwo:

$$A>B,\ A>C,\ B>C,\ B>D,\ C>B,\ C>D,\ E>F$$

- Przyczynowość:

$$A \rightarrow B,\ A \rightarrow C,\ B \rightarrow D,\ C \rightarrow D,\ E \rightarrow F$$

- Współbieżność:

$$B||C,\ C||B$$

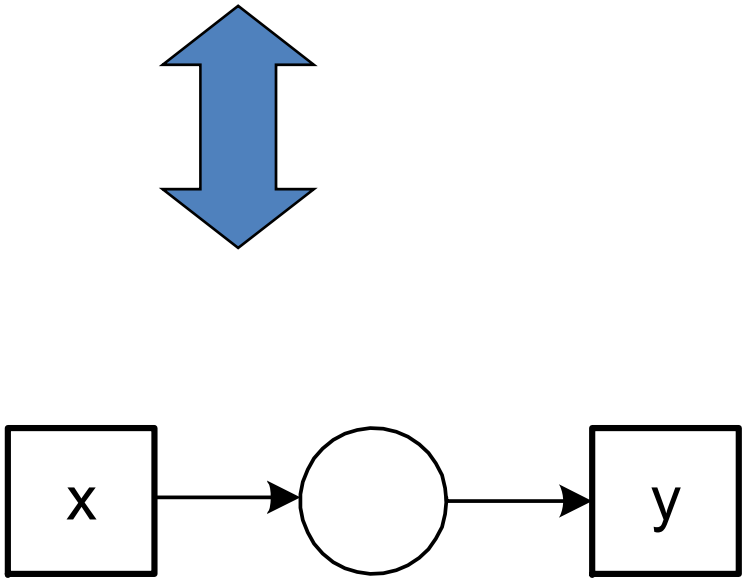
- Wybór:

$$E\#A,\ E\#B,\ E\#C,\ E\#D,\ E\#A,\ E\#B,\ E\#V,\ E\#D$$

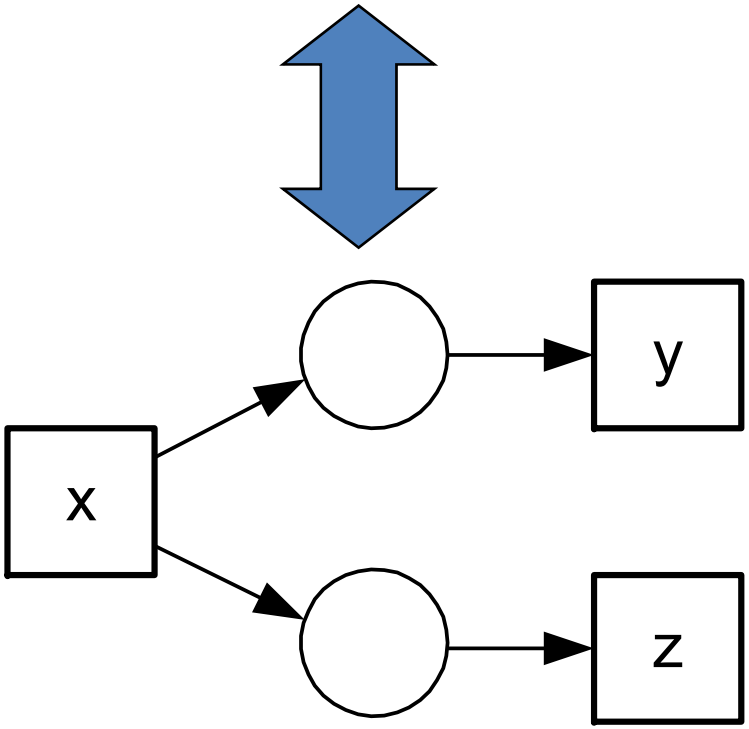
- **ABCD**
- **ACBD**
- **EF**

Algorytm α - reguły transformacji

a) $x \rightarrow y$

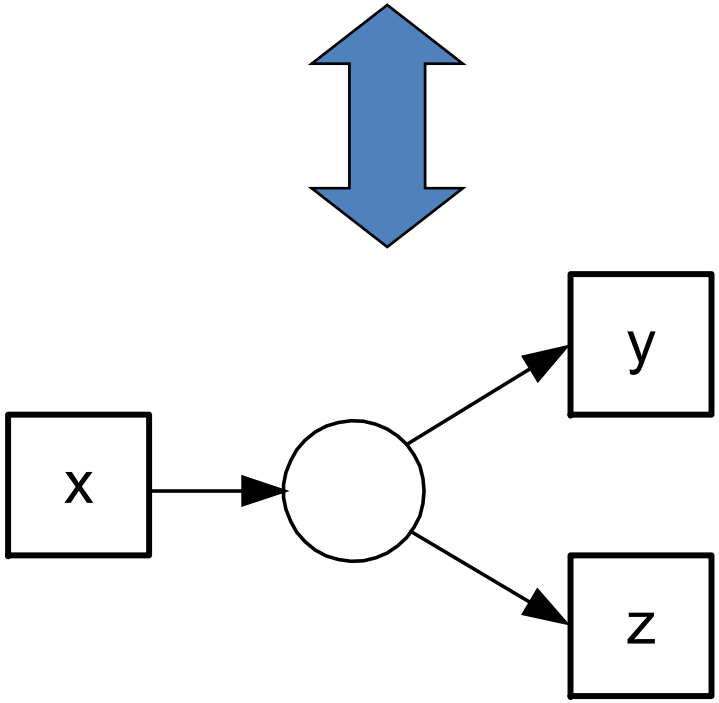


b) $x \rightarrow y$, $x \rightarrow z$, and $y \| z$

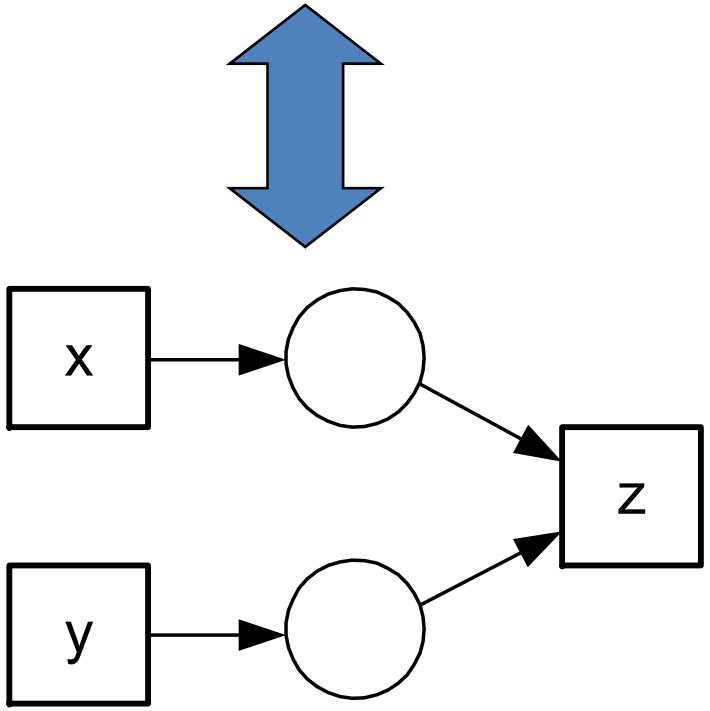


Algorytm α - reguły transformacji

c) $x \rightarrow y$, $x \rightarrow z$, and $y \# z$

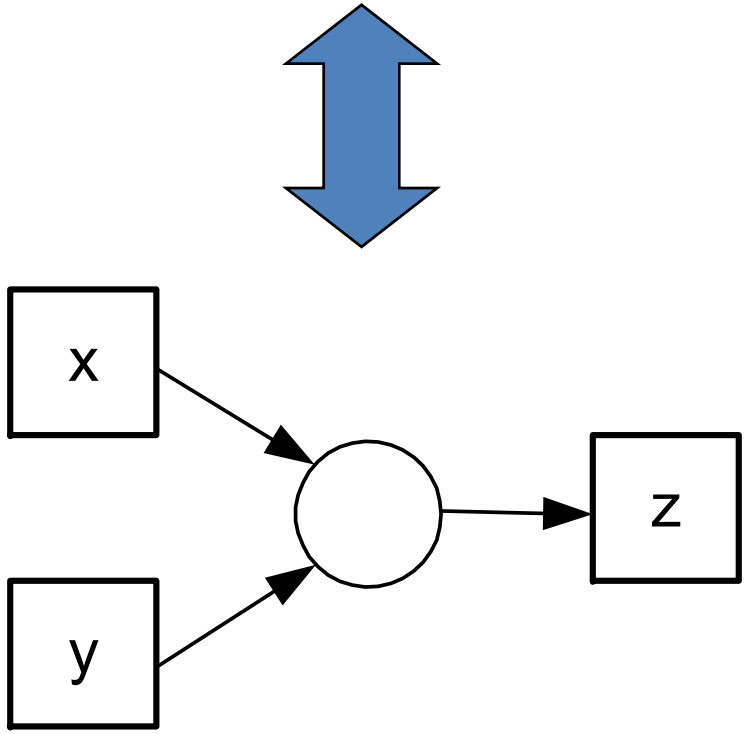


d) $x \rightarrow z$, $y \rightarrow z$, and $x \| y$



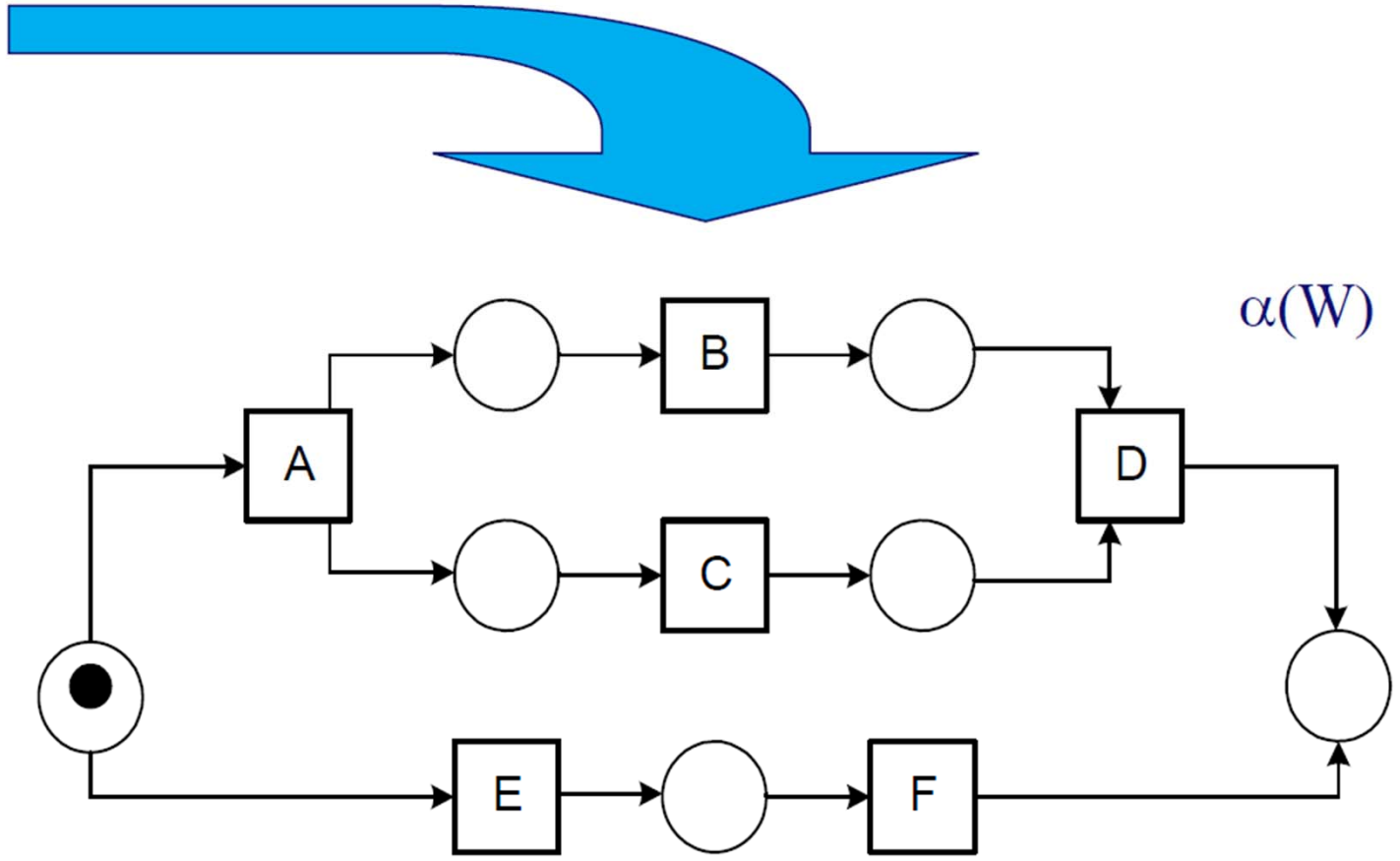
Algorytm α - reguły transformacji

e) $x \rightarrow z$, $y \rightarrow z$, and $x \# y$



Case	Task
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task E
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task F
case 4	task D

Wynik odkrywania modelu



Formalna definicja algorytmu α

Niech W będzie logiem procesu zdefiniowanym na zbiorze elementarnych zadań T.

α(W) jest zdefiniowane następująco:

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\, a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\, b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,
7. $F_W = \{ (a,p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)},b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W,t) \mid t \in T_I \} \cup \{ (t,o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W, T_W, F_W)$.

Sieć socjalna łącząca wykonawców

Sieć socjalna zbudowana na podstawie przekazywania pracy w ramach procesów