Algebraiczne podejście do reprezentacji procesów

Przemysław Szymoniak

2025

Spis treści

Abstrakt

Celem niniejszej pracy jest przedstawienie formalnej struktury zwanej procesem algebraicznym, która umożliwia algebraiczny opis procesów wymiany informacji. Proces algebraiczny definiuje obiekty, ich stany, relacje oraz funkcje przejścia, co ułatwia formalne modelowanie i analizę procesów, takich jak przepływy materiałów, alokacje zasobów oraz interakcje systemowe. Podejście to otwiera drogę do badania rodzin procesów oraz ich własności algebraicznych i topologicznych.

1 Wprowadzenie do Procesu Algebraicznego

Koncepcja procesu algebraicznego inspirowana jest potrzebą formalizacji procesów w których informacje, zasoby lub materiały przepływają pomiędzy wzajemnie połączonymi komponentami. Ramy te mają na celu zapewnienie uogólnionej reprezentacji procesów przy użyciu matematyki. Precyzyjne zdefiniowanie obiektów, ich interakcji oraz sposobu, w jaki przesyłają między sobą informacje, czyni proces algebraiczny podstawą do analizy własności i zachowań złożonych procesów w różnych zastosowaniach.

1.1 Proces Algebraiczny

Proces algebraiczny $\mathcal{P}$ zdefiniowany jest jako trójka:

$$\mathcal{P} = (\mathcal{O}, \mathcal{R}, \mathcal{T}), \tag{1}$$

gdzie:

- $\mathcal{O}$ — zbiór obiektów procesu, $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$,
- $\mathcal{R}$ — zbiór relacji między obiektami, $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{O}$,
- $\mathcal{T}$ — zbiór funkcji przejścia opisujący charakter relacji, $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$.

Z uwagi na ogólność powyższej definicji, wprowadzone zostaną warunki ograniczające strukturę $\mathcal{P}$.

1.2 Warunki Spójności

Ograniczenia te definiują podstawowe własności procesu algebraicznego, które eliminują redundancje oraz zapewniają wewnętrzną spójność przepływu informacji w procesie. Mają one na celu precyzyjne zdefiniowanie struktury procesu i jego funkcjonalności.

Warunek I: Zupełność procesu: Proces algebraiczny jest zupełny, jeżeli każdy obiekt uczestniczy w co najmniej jednej relacji:

$$\forall o_i \in \mathcal{O},\ \exists (o_i, o_j) \in \mathcal{R} \lor (o_j, o_i) \in \mathcal{R}. \tag{2}$$

Obiekty które nie uczestniczą w żadnej relacji są zbędne, ponieważ nie wpływają na proces. Zapewnienie, że każdy obiekt jest częścią co najmniej jednej relacji, gwarantuje, że wszystkie elementy $\mathcal{P}$ aktywnie uczestniczą w przepływie procesu.

Warunek II: Spójność przepływu: Proces algebraiczny jest spójny, jeżeli każda relacja opisana jest przez odpowiednią funkcje przejścia:

$$\forall (o_i, o_j) \in \mathcal{R},\ \exists T_{i,j} \in \mathcal{T}. \tag{3}$$

Relacja bez przypisanej funkcji przejścia oznacza, że sposób interakcji między obiektami jest niezdefiniowany, co prowadzi do przerwania przepływu procesu. Powiązanie każdej relacji z funkcją przejścia zapewnia, że interakcje w procesie są dobrze określone.

Dobra określoność: Proces algebraiczny jest dobrze określony, jeżeli jest zupełny i spójny:

$$\mathcal{P} \text{ jest określony} \iff \text{Zupełność procesu} \wedge \text{Spójność przepływu}. \tag{4}$$

Warunki (2) i (3) ustanawiają dobrą określoność procesu algebraicznego, zapewniając, że wszystkie obiekty, relacje i funkcje przejścia są ze sobą powiązane oraz dobrze zdefiniowane. Gwarantują, że graf (topologia procesu) nie zawiera izolowanych węzłów (nieprzypisanych obiektów) ani krawędzi bez określonego kierunku (brak funkcji przejścia). W przeciwnym przypadku proces nazywamy nieokreślonym.

1.3 Obiekty $\mathcal{O}$

W kontekście procesu algebraicznego, obiekt $o_i \in \mathcal{O}$ reprezentuje odrębną jednostkę w procesie, charakteryzowaną przez zbiór parametrów definiujących jego rolę i właściwości:

$$o_i = \{p_1, p_2, \ldots, p_n\}, \tag{5}$$

gdzie:

- p_k to parametry obiektu, takie jak właściwości materiałowe, ograniczenia operacyjne lub inne istotne atrybuty,
- każdy parametr p_k ma przypisaną określoną wartość, która odzwierciedla stan lub konfigurację obiektu w czasie dla zdefiniowanego procesu.

Właściwości Obiekty o_i w procesie algebraicznym charakteryzują się następującymi właściwościami:

1. **Złożoność:** Każdy obiekt o_i jest reprezentowany jako zbiór parametrów $\{p_1, p_2, \ldots, p_n\}$, które opisują jego stan oraz rolę w procesie. Liczba i typ parametrów mogą się różnić w zależności od charakteru obiektu.
2. **Stan:** Obiekty posiadają przestrzeń stanów S_i, która określa wszystkie możliwe konfiguracje parametrów p_k. Stan $s_i \in S_i$ w dowolnym czasie t definiuje aktualną konfigurację obiektu.
3. **Autonomia:** Obiekty mogą operować autonomicznie, podejmując decyzje na podstawie swoich parametrów, lub być całkowicie zależne od danych wejściowych I_i wynikających z relacji $\mathcal{R}$.

Rola Obiekty są fundamentalnymi składnikami procesu algebraicznego $\mathcal{P}$ i stanowią podstawowe elementy konstrukcyjne dla:

- **Definiowania relacji:** Obiekty są połączone za pomocą relacji $\mathcal{R}$, które opisują przepływ informacji, zasobów lub interakcje między nimi.
- **Regulowania funkcji przejścia:** Dynamika procesu jest określana przez funkcje przejścia $\mathcal{T}$, które operują na parametrach obiektów.

Przykłady Obiekty w procesie algebraicznym mogą reprezentować różnorodne jednostki w zależności od dziedziny zastosowania. Na przykład:

- **Produkcja:** Maszyna charakteryzowana przez parametry, takie jak czas cyklu, wydajność produkcji i zużycie energii.
- **Logistyka:** Magazyn zdefiniowany przez swoją pojemność i aktualny stan zapełnienia.
- **Systemy obliczeniowe:** Serwer opisany atrybutami, takimi jak moc obliczeniowa i dostępna pamięć.

Te przykłady podkreślają ogólność definicji obiektu, która może być dostosowana do wymagań konkretnego procesu modelowanego w danej dziedzinie.

1.4 Relacje $\mathcal{R}$

Relacje $\mathcal{R}$ opisują interakcje, zależności oraz przepływy informacji lub materiałów między obiektami w procesie algebraicznym. Formalnie, zbiór relacji definiuje się jako:

$$\mathcal{R} = \{(o_i, o_j) \mid o_i \to o_j\}, \tag{6}$$

gdzie:

- $o_i, o_j \in \mathcal{O}$ są obiektami w procesie,
- $o_i \to o_j$ oznacza, że obiekt o_i dostarcza dane wejściowe do obiektu o_j, co reprezentuje relację skierowaną.

Właściwości

- **Skierowany charakter:** Relacje w $\mathcal{R}$ są skierowane, co oznacza, że $(o_i, o_j) \neq (o_j, o_i)$. Zapewnia to jednoznaczne określenie źródła i celu przepływu informacji lub materiałów.
- **Wielość:** Obiekt o_i może mieć wiele wychodzących lub wchodzących relacji:

$$\{(o_i, o_k), (o_i, o_l), \dots\} \quad \text{lub} \quad \{(o_k, o_i), (o_l, o_i), \dots\}. \tag{7}$$

- **Relacje z samym sobą:** Dopuszczalne są relacje, w których obiekt o_i jest jednocześnie źródłem i celem:

$$(o_i, o_i) \in \mathcal{R}. \tag{8}$$

  Tego typu relacje mogą modelować procesy wewnętrzne lub samoaktualizację stanu obiektu.

Rola Relacje $\mathcal{R}$ odgrywają kluczową rolę w procesie algebraicznym, ponieważ:

- **Definiują interakcje:** Relacje określają, w jaki sposób obiekty oddziałują na siebie, czy to poprzez bezpośredni transfer danych, czy zależności zasobowe.
- **Kierują przejściami:** Funkcje przejścia $\mathcal{T}$ działają na podstawie relacji w $\mathcal{R}$, co zapewnia ewolucję procesu zgodnie z określoną strukturą.
- **Określają topologię:** Zbiór $\mathcal{R}$ definiuje strukturę topologiczną procesu, np. czy tworzy liniowy łańcuch, drzewo czy sieć.

Przykłady Relacje mogą reprezentować różne typy interakcji w zależności od charakteru procesu. Poniżej przedstawiono przykłady typów relacji w różnych procesach:

- **Przepływ informacji:** $o_i \to o_j$ oznacza transfer danych lub instrukcji z obiektu o_i do obiektu o_j. *Przykład:* W procesie obliczeniowym:

$$\mathcal{R} = \{(o_{\text{serwer}}, o_{\text{klient}}), (o_{\text{baza\_danych}}, o_{\text{serwer}})\}. \tag{9}$$

- **Przepływ materiałów:** $o_i \to o_j$ wskazuje na transfer zasobów fizycznych lub komponentów. *Przykład:* W procesie produkcyjnym:

$$\mathcal{R} = \{(o_{\text{magazyn}}, o_{\text{maszyna}}), (o_{\text{maszyna}}, o_{\text{bufor\_wyjściowy}})\}. \tag{10}$$

- **Zależność:** $o_i \to o_j$ odzwierciedla, że działanie lub stan obiektu o_j zależy od obiektu o_i. *Przykład:* W harmonogramie projektu:

$$\mathcal{R} = \{(o_{\text{zadanie1}}, o_{\text{zadanie2}}), (o_{\text{zadanie2}}, o_{\text{zadanie3}})\}. \tag{11}$$

1.5 Funkcje Przejścia $\mathcal{T}$

Zbiór funkcji przejścia $\mathcal{T}$ definiuje dynamikę procesu algebraicznego, określając, w jaki sposób obiekty ewoluują i wchodzą w interakcje w czasie. Funkcje te operują na parametrach obiektów i ich wejściach:

$$\mathcal{T} = \{T_1, T_2, \dots, T_m\}, \tag{12}$$

gdzie:

- T_k reprezentuje konkretną funkcję przejścia opisującą interakcję lub transformację w procesie.

Każda funkcja przejścia T_k jest zdefiniowana jako odwzorowanie:

$$T_k : S_i \times I_i \to S_i, \tag{13}$$

gdzie:

- S_i jest przestrzenią stanów obiektu o_i, reprezentującą jego wewnętrzne parametry,
- I_i jest przestrzenią wejść obiektu, wyznaczoną przez jego relacje $\mathcal{R}$, reprezentującą zewnętrzne wpływy lub zależności.

Właściwości

- **Deterministyczność i stochastyczność:** Funkcje przejścia mogą być deterministyczne, generując jednoznaczny wynik dla danego wejścia, lub stochastyczne, uwzględniające losowe zmienności.
- **Dziedzina definicji:** Każda funkcja T_k jest zdefiniowana dla określonych obiektów i wejść, co zapewnia jej działanie w odpowiednim zakresie procesu.
- **Ciągłość lub dyskretność:** W zależności od charakteru procesu, T_k może reprezentować ciągłe transformacje lub dyskretne zmiany stanów.

Rola Funkcje przejścia $\mathcal{T}$ stanowią fundament procesu algebraicznego, umożliwiając:

- **Dynamikę systemu:** Określenie, w jaki sposób stany obiektów zmieniają się w czasie pod wpływem interakcji i wejść.
- **Interakcje:** Ujęcie zależności i przepływów określonych przez relacje $\mathcal{R}$.

Przykłady

Transfer materiałów w systemie produkcyjnym:

$$T_{\text{transfer}}(s_i, i) = s_i - i, \tag{14}$$

gdzie:

- s_i jest aktualnym poziomem materiału w obiekcie,
- i to ilość materiału przekazana do innego obiektu.

Przetwarzanie danych w systemie obliczeniowym:

$$T_{\text{process}}(s_i, i) = s_i + f(i), \tag{15}$$

gdzie:

- s_i to aktualnie przetworzone dane,
- i to dane wejściowe,
- $f(i)$ reprezentuje funkcję przetwarzającą dane wejściowe.

Zużycie energii w maszynie:

$$T_{\text{energy}}(s_i, t) = s_i - (p \cdot t), \tag{16}$$

gdzie:

- s_i to aktualny poziom energii,
- p to tempo zużycia energii,
- t to czas trwania.

2 Równoważność Procesów Algebraicznych

Równoważność procesów algebraicznych $\mathcal{P} = (\mathcal{O}, \mathcal{R}, \mathcal{T})$ oraz $\mathcal{P}' = (\mathcal{O}', \mathcal{R}', \mathcal{T}')$ można definiować na różne sposoby, uwzględniając różne aspekty ich komponentów. Niniejsza sekcja szczegółowo omawia hierarchię kryteriów równoważności dla elementów procesu algebraicznego.

2.1 Kryteria równoważności zbiorów obiektów

Równoważność obiektów $\mathcal{O}$ i $\mathcal{O}'$ można rozpatrywać w następujących krokach od najsłabszego do najsilniejszego kryterium:

1. **Równoważność mocy zbiorów** Pierwszym kryterium równoważności jest porównanie mocy zbiorów $\mathcal{O}$ i $\mathcal{O}'$. Zbiory są równoważne w sensie ich mocy, jeśli zawierają tę samą liczbę obiektów:

$$\mathcal{O} \equiv_{\text{moc}} \mathcal{O}' \iff |\mathcal{O}| = |\mathcal{O}'|, \tag{17}$$

gdzie $|\mathcal{O}|$ oznacza liczbę obiektów w zbiorze. Kryterium to zapewnia, że w procesach algebraicznych $\mathcal{P}$ i $\mathcal{P}'$ uczestniczy ta sama liczba obiektów. Spełnienie tego warunku jest konieczne do spełnienia pozostałych kryteriów.

2. **Równoważność wymiarów przestrzeni stanów** Drugim kryterium równoważności jest zgodność wymiarów przestrzeni stanów obiektów w zbiorach $\mathcal{O}$ i $\mathcal{O}'$. Zbiory są równoważne w sensie wymiarów przestrzeni stanów, jeśli przestrzenie stanów odpowiadających obiektów mają tę samą liczbę wymiarów:

$$\mathcal{O} \equiv_{\text{wymiar}} \mathcal{O}' \iff \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \dim(S_i) = \dim(S_i'), \tag{18}$$

gdzie $\dim(S_i)$ oznacza wymiar przestrzeni stanów S_i. To kryterium gwarantuje, że każdy obiekt w $\mathcal{O}$ ma odpowiadający sobie obiekt w $\mathcal{O}'$ o tej samej liczbie parametrów. Spełnienie tego warunku implikuje spełnienie pierwszego kryterium.

Dowód implikacji $\mathcal{O} \equiv_{\textbf{wymiar}} \mathcal{O}' \implies \mathcal{O} \equiv_{\textbf{moc}} \mathcal{O}'$ Załóżmy, że zbiory obiektów $\mathcal{O}$ i $\mathcal{O}'$ są równoważne w sensie wymiarów przestrzeni stanów:

$$\mathcal{O} \equiv_{\text{wymiar}} \mathcal{O}' \iff \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \dim(S_i) = \dim(S_i'). \tag{19}$$

Z definicji równoważności wymiarów przestrzeni stanów wynika, że dla każdego obiektu $o_i \in \mathcal{O}$ istnieje dokładnie jeden obiekt $o_i' \in \mathcal{O}'$, który ma identyczny wymiar przestrzeni stanów $\dim(S_i)$. Oznacza to, że liczba obiektów w zbiorze $\mathcal{O}$ jest taka sama jak liczba obiektów w zbiorze $\mathcal{O}'$.

Formalnie:

$$\forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' \quad \text{oraz} \quad \forall o_i' \in \mathcal{O}', \exists o_i \in \mathcal{O}. \tag{20}$$

Z powyższego wynika, że istnieje bijekcja $\phi : \mathcal{O} \to \mathcal{O}'$, co oznacza, że:

$$|\mathcal{O}| = |\mathcal{O}'|. \tag{21}$$

Stąd:

$$\mathcal{O} \equiv_{\text{wymiar}} \mathcal{O}' \implies \mathcal{O} \equiv_{\text{moc}} \mathcal{O}'. \tag{22}$$

c.n.d.

3. **Równoważność dziedzin parametrów** Trzecim kryterium jest zgodność dziedzin parametrów w przestrzeniach stanów. Zbiory są równoważne w sensie dziedzin parametrów, jeśli odpowiadające sobie przestrzenie stanów mają identyczne dziedziny parametrów:

$$\mathcal{O} \equiv_{\text{dziedzina}} \mathcal{O}' \iff \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \forall p_j \in S_i,\ \text{dom}(p_j) = \text{dom}(p_j'), \tag{23}$$

gdzie $\text{dom}(p_j)$ oznacza dziedzinę parametru p_j. To kryterium zakłada identyczność przestrzeni stanów odpowiadających sobie obiektów, co oznacza, że parametry mają takie same dziedziny. Spełnienie tego kryterium automatycznie implikuje spełnienie kryteriów pierwszego i drugiego.

Dowód implikacji $\mathcal{O} \equiv_{\textbf{dziedzina}} \mathcal{O}' \implies \mathcal{O} \equiv_{\textbf{wymiar}} \mathcal{O}'$ Załóżmy, że zbiory obiektów $\mathcal{O}$ i $\mathcal{O}'$ są równoważne w sensie dziedzin parametrów:

$$\mathcal{O} \equiv_{\text{dziedzina}} \mathcal{O}' \iff \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \forall p_j \in S_i,\ \text{dom}(p_j) = \text{dom}(p_j'). \tag{24}$$

Z powyższego założenia wynika, że dla każdego obiektu $o_i \in \mathcal{O}$ istnieje dokładnie jeden obiekt $o_i' \in \mathcal{O}'$, którego parametry mają identyczne dziedziny. Liczba parametrów w przestrzeni stanów S_i jest równa liczbie parametrów w przestrzeni stanów S_i', ponieważ każda dziedzina $\text{dom}(p_j)$ odpowiada jednemu parametrowi p_j.

Formalnie:

$$\forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : |S_i| = |S_i'| \implies \dim(S_i) = \dim(S_i'). \tag{25}$$

Analogicznie:

$$\forall o_i' \in \mathcal{O}', \exists o_i \in \mathcal{O} : |S_i| = |S_i'| \implies \dim(S_i') = \dim(S_i). \tag{26}$$

Stąd wynika, że przestrzenie stanów odpowiadających sobie obiektów o_i i o_i' mają identyczne wymiary:

$$\dim(S_i) = \dim(S_i'). \tag{27}$$

Zatem:

$$\mathcal{O} \equiv_{\text{dziedzina}} \mathcal{O}' \implies \mathcal{O} \equiv_{\text{wymiar}} \mathcal{O}'. \tag{28}$$

c.n.d.

4. **Równoważność wartości parametrów** Czwartym kryterium jest zgodność wartości parametrów ustalonych w procesie algebraicznym. W przeciwieństwie do równoważności na podstawie dziedzin parametrów, która zapewnia identyczność przestrzeni stanów, równoważność na podstawie wartości parametrów dotyczy konkretnych ustalonych wartości parametrów opisujących obiekty. Zbiory są równoważne w sensie wartości parametrów, jeśli odpowiadające sobie parametry w procesie algebraicznym mają identyczne wartości:

$$\mathcal{O} \equiv_{\text{parametry}} \mathcal{O}' \iff \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \forall p_j \in s_i,\ p_j = p_j', \tag{29}$$

gdzie p_j oraz p_j' są ustalonymi parametrami stanów s_i i s_i'.

Równoważność wartości parametrów sprawia, że obiekty w zbiorach $\mathcal{O}$ i $\mathcal{O}'$ są skonfigurowane w sposób identyczny. Jest to najsilniejsze kryterium, które wymaga, aby odpowiadające sobie parametry w obu zbiorach miały identyczne wartości. Spełnienie tego kryterium implikuje spełnienie pierwszego i drugiego kryterium, lecz niekoniecznie trzeciego, ponieważ wartości parametrów mogą być równe pomimo różnic w ich dziedzinach.

Dowód implikacji $\mathcal{O} \equiv_{\textbf{parametry}} \mathcal{O}' \implies \mathcal{O} \equiv_{\textbf{wymiar}} \mathcal{O}'$ Załóżmy, że:

$$\mathcal{O} \equiv_{\text{parametry}} \mathcal{O}'. \tag{30}$$

Oznacza to, że:

$$\forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \forall p_j \in s_i,\ p_j = p_j'. \tag{31}$$

Z warunku $p_j = p_j'$ wynika, że zarówno S_i, jak i S_i' mają tę samą liczbę parametrów, ponieważ każda wartość p_j w S_i odpowiada dokładnie jednej wartości p_j' w S_i'. Formalnie:

$$\dim(S_i) = \dim(S_i'), \quad \forall o_i \in \mathcal{O}. \tag{32}$$

Zatem:

$$\mathcal{O} \equiv_{\text{parametry}} \mathcal{O}' \implies \mathcal{O} \equiv_{\text{wymiar}} \mathcal{O}'. \tag{33}$$

c.n.d

Dowód implikacji $\mathcal{O} \equiv_{\textbf{parametry}} \mathcal{O}' \implies \mathcal{O} \equiv_{\textbf{moc}} \mathcal{O}'$ Załóżmy ponownie, że:

$$\mathcal{O} \equiv_{\text{parametry}} \mathcal{O}'. \tag{34}$$

Oznacza to, że istnieje odwzorowanie między zbiorami $\mathcal{O}$ i $\mathcal{O}'$, takie że dla każdego obiektu $o_i \in \mathcal{O}$ istnieje dokładnie jeden odpowiadający obiekt $o_i' \in \mathcal{O}'$. W szczególności liczba obiektów w zbiorach $\mathcal{O}$ i $\mathcal{O}'$ musi być równa:

$$|\mathcal{O}| = |\mathcal{O}'|. \tag{35}$$

Zatem:

$$\mathcal{O} \equiv_{\text{parametry}} \mathcal{O}' \implies \mathcal{O} \equiv_{\text{moc}} \mathcal{O}'. \tag{36}$$

c.n.d

Dowód implikacji $\mathcal{O} \equiv_{\textbf{parametry}} \mathcal{O}' \not\Longrightarrow \mathcal{O} \equiv_{\textbf{dziedzina}} \mathcal{O}'$ Załóżmy, że zbiory obiektów $\mathcal{O}$ i $\mathcal{O}'$ są równoważne w sensie wartości parametrów:

$$\mathcal{O} \equiv_{\text{parametry}} \mathcal{O}' \iff \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \forall p_j \in s_i,\ p_j = p_j'. \tag{37}$$

Oznacza to, że każdy parametr stanu $p_j \in s_i$ ma wartość równą wartości parametru stanu $p_j' \in s_i'$. Jednak równość wartości parametrów nie gwarantuje, że ich dziedziny są identyczne. Rozważmy przykład:

- $p_1 = p_1' = 5$,
- ale $\mathrm{dom}(p_1) = \{1, 2, 3, 4, 5\}$ i $\mathrm{dom}(p_1') = \{5, 6, 7, 8, 9\}$.

W takim przypadku:

$$\mathrm{dom}(p_1) \neq \mathrm{dom}(p_1'), \tag{38}$$

co oznacza, że:

$$\mathcal{O} \equiv_{\text{parametry}} \mathcal{O}' \not\Longrightarrow \mathcal{O} \equiv_{\text{dziedzina}} \mathcal{O}'. \tag{39}$$

c.n.d

Dowód implikacji $\mathcal{O} \equiv_{\textbf{dziedzina}} \mathcal{O}' \not\Longrightarrow \mathcal{O} \equiv_{\textbf{parametry}} \mathcal{O}'$ Załóżmy, że zbiory obiektów $\mathcal{O}$ i $\mathcal{O}'$ są równoważne w sensie dziedzin parametrów:

$$\mathcal{O} \equiv_{\text{dziedzina}} \mathcal{O}' \iff \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \forall p_j \in S_i,\ \operatorname{dom}(p_j) = \operatorname{dom}(p_j'). \tag{40}$$

Oznacza to, że każdy parametr $p_j \in S_i$ ma identyczną dziedzinę co odpowiadający mu parametr $p_j' \in S_i'$. Jednak równość dziedzin parametrów nie gwarantuje, że wartości parametrów są takie same. Rozważmy przykład:

- $\operatorname{dom}(p_1) = \operatorname{dom}(p_1') = \{1, 2, 3, 4, 5\}$,
- ale $p_1 = 3$ i $p_1' = 5$.

W takim przypadku:

$$p_1 \neq p_1', \tag{41}$$

co oznacza, że:

$$\mathcal{O} \equiv_{\text{dziedzina}} \mathcal{O}' \not\Longrightarrow \mathcal{O} \equiv_{\text{parametry}} \mathcal{O}'. \tag{42}$$

c.n.d

Z dwóch powyższych dowodów wynika, że:

$$\mathcal{O} \equiv_{\text{parametry}} \mathcal{O}' \not\Longleftrightarrow \mathcal{O} \equiv_{\text{dziedzina}} \mathcal{O}'. \tag{43}$$

Oba kryteria są niezależne i spełnienie jednego z nich nie gwarantuje spełnienia drugiego.

5. **Równoważność zbiorów obiektów** Kryteria równoważności są uporządkowane hierarchicznie, gdzie kryteria pierwsze i drugie stanowią fundament, natomiast kryteria trzecie i czwarte pozwalają na bardziej szczegółową analizę. Kryterium równoważności dziedzin parametrów oraz równoważności wartości parametrów są niezależne, co oznacza, że ich spełnienie nie wynika automatycznie jedno z drugiego.

   Zbiory obiektów $\mathcal{O}$ i $\mathcal{O}'$ są równoważne, jeśli spełnione są następujące warunki:

$$\mathcal{O} \equiv \mathcal{O}' \iff \begin{cases} \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \forall p_j \in S_i,\ \operatorname{dom}(p_j) = \operatorname{dom}(p_j'), \\ \forall o_i \in \mathcal{O}, \exists o_i' \in \mathcal{O}' : \forall p_j \in s_i,\ p_j = p_j'. \end{cases} \tag{44}$$

   Ta definicja uwzględnia zarówno zgodność dziedzin parametrów, jak i zgodność ich wartości, co wystarcza do zapewnienia pełnej równoważności zbiorów obiektów. Warunki mocy zbiorów oraz wymiarów przestrzeni stanów wynikają logicznie z tych kryteriów i nie muszą być explicite uwzględniane.

2.2 Kryteria równoważności zbiorów relacji

1. **Warunek konieczny równoważności zbiorów relacji** W celu zdefiniowania równoważności zbiorów relacji procesów algebraicznych $\mathcal{P}$ i $\mathcal{P}'$ niezbędne jest spełnienie kryterium równoważności mocy zbiorów obiektów (20).

   Dowód implikacji $\mathcal{O} \not\equiv_{\textbf{moc}} \mathcal{O}' \implies \mathcal{R} \not\equiv \mathcal{R}'$ Załóżmy, że zbiory obiektów $\mathcal{O}$ i $\mathcal{O}'$ procesów algebraicznych $\mathcal{P}$ oraz $\mathcal{P}'$ nie są równoważne w sensie mocy:

$$\mathcal{O} \not\equiv_{\text{moc}} \mathcal{O}' \iff |\mathcal{O}| \neq |\mathcal{O}'|.$$

   Z warunku spójności (2) wynika, że każdy obiekt $o_i \in \mathcal{O}$ musi uczestniczyć w co najmniej jednej relacji:

$$\forall o_i \in \mathcal{O},\ \exists (o_i, o_j) \in \mathcal{R} \vee (o_j, o_i) \in \mathcal{R}.$$

   Analogicznie:

$$\forall o_i' \in \mathcal{O}',\ \exists (o_i', o_j') \in \mathcal{R}' \vee (o_j', o_i') \in \mathcal{R}'.$$

Jeśli $|\mathcal{O}| \neq |\mathcal{O}'|$, to oznacza, że jeden z procesów algebraicznych ma więcej obiektów niż drugi:

$$|\mathcal{O}| > |\mathcal{O}'| \quad \text{lub} \quad |\mathcal{O}| < |\mathcal{O}'|.$$

Załóżmy bez utraty ogólności, że $|\mathcal{O}| > |\mathcal{O}'|$. W takim przypadku istnieje co najmniej jeden obiekt $o_i \in \mathcal{O}$, który nie ma odpowiednika w $\mathcal{O}'$. Dla tego obiektu z warunku spójności wynika, że uczestniczy on w co najmniej jednej relacji w $\mathcal{R}$:

$$\exists o_i \in \mathcal{O}, \nexists o_i' \in \mathcal{O}' \implies \exists (o_i, o_j) \in \mathcal{R} \vee (o_j, o_i) \in \mathcal{R}.$$

Ponieważ brak odpowiednika dla o_i w $\mathcal{O}'$ oznacza, że o_i nie może uczestniczyć w żadnej relacji w $\mathcal{R}'$, nie jest możliwe zdefiniowanie pełnej równoważności relacji $\mathcal{R} \equiv \mathcal{R}'$.

Należy zauważyć, że równość mocy zbiorów relacji $|\mathcal{R}| = |\mathcal{R}'|$ może zajść, jeśli $\mathcal{R}'$ zawiera więcej relacji dla pojedynczych obiektów niż $\mathcal{R}$. Jednakże taka sytuacja prowadzi do braku zgodności w parowaniu relacji między obiektami, co również wyklucza pełną równoważność relacji $\mathcal{R}$ i $\mathcal{R}'$.

Podsumowując:

$$\mathcal{O} \not\equiv_{\text{moc}} \mathcal{O}' \implies \mathcal{R} \not\equiv \mathcal{R}'.$$

c.n.d.

2. **Równoważność mnogościowa** Równoważność zbiorów relacji $\mathcal{R}$ i $\mathcal{R}'$ można definiować w sposób niezależny od szczegółowych właściwości obiektów, uwzględniając jedynie strukturę zbiorów relacji. Równoważność mnogościowa relacji jest zdefiniowana jako istnienie bijekcji między elementami zbiorów relacji:

$$\mathcal{R} \equiv_{\text{mnogościowa}} \mathcal{R}' \iff \exists \phi : \mathcal{R} \to \mathcal{R}', \quad \forall (o_i, o_j) \in \mathcal{R},\ \phi((o_i, o_j)) = (o_i', o_j'). \tag{45}$$

gdzie:

- ϕ jest bijekcją, co oznacza, że każda relacja $(o_i, o_j) \in \mathcal{R}$ ma dokładnie jednego odpowiednika $(o_i', o_j') \in \mathcal{R}'$, a każda relacja w $\mathcal{R}'$ ma dokładnie jednego prekursora w $\mathcal{R}$.
- Obiekty o_i, o_j, o_i', o_j' są traktowane wyłącznie jako elementy relacji, bez dodatkowych założeń co do ich charakterystyki.

Definicja ta opiera się wyłącznie na strukturze zbiorów relacji i pozwala na porównywanie procesów algebraicznych, dla których jedynym wspólnym warunkiem jest równoważność mocy zbiorów obiektów.

2.3 Kryteria równoważności zbiorów funkcji przejścia

1. **Warunek konieczny równoważności funkcji przejścia:**

Aby zdefiniować równoważność zbiorów funkcji przejścia $\mathcal{T}$ i $\mathcal{T}'$, procesy algebraiczne $\mathcal{P} = (\mathcal{O}, \mathcal{R}, \mathcal{T})$ oraz $\mathcal{P}' = (\mathcal{O}', \mathcal{R}', \mathcal{T}')$ muszą być równoważne w sensie relacji:

$$\mathcal{R} \equiv \mathcal{R}'.$$

Dowód: Z warunku spójności procesu algebraicznego wynika, że każda relacja w $\mathcal{R}$ musi być opisana odpowiednią funkcją przejścia:

$$\forall (o_i, o_j) \in \mathcal{R}, \exists T_{ij} \in \mathcal{T}.$$

Analogicznie:

$$\forall (o_i', o_j') \in \mathcal{R}', \exists T_{ij}' \in \mathcal{T}'.$$

Jeśli $\mathcal{R} \not\equiv \mathcal{R}'$, to zachodzi co najmniej jedna z dwóch sytuacji:

- Istnieje relacja $(o_i, o_j) \in \mathcal{R}$, dla której brak odpowiednika w $\mathcal{R}'$. Odpowiadająca funkcja przejścia $T_{ij} \in \mathcal{T}$ nie ma odpowiednika w $\mathcal{T}'$.

- Istnieje relacja $(o'_i, o'_j) \in \mathcal{R}'$, dla której brak odpowiednika w $\mathcal{R}$. Odpowiadająca funkcja przejścia $T'_{ij} \in \mathcal{T}'$ nie ma odpowiednika w $\mathcal{T}$.

W obu przypadkach porównanie $\mathcal{T}$ i $\mathcal{T}'$ jest niemożliwe, ponieważ zestawy funkcji przejścia odnoszą się do różnych struktur relacji. Stąd:

$$\mathcal{R} \equiv \mathcal{R}' \quad \text{jest warunkiem koniecznym dla} \quad \mathcal{T} \equiv \mathcal{T}'.$$

2. **Dziedzina i przeciwdziedzina funkcji przejścia**
Dziedzina funkcji przejścia T_k jest produktem kartezjańskim dziedzin parametrów obiektów, które są powiązane przez relację:

$$\mathrm{dom}(T_k) = \mathrm{dom}(p_{j_1}) \times \mathrm{dom}(p_{j_2}) \times \cdots \times \mathrm{dom}(p_{j_n}), \tag{46}$$

gdzie:

- $\mathrm{dom}(p_{j_m})$ to dziedzina m-tego parametru p_{j_m} w przestrzeni stanu obiektu o_i,
- n to liczba parametrów wchodzących do dziedziny funkcji przejścia T_k.

Funkcja przejścia T_k przekształca dziedzinę $\mathrm{dom}(T_k)$ w przeciwdziedzinę $\mathrm{codom}(T_k)$, która jednocześnie jest podprzestrzenią przestrzeni stanu drugiego obiektu, do którego odnosi się funkcja:

$$\mathrm{codom}(T_k) \subseteq \mathrm{dom}(p'_{j_1}) \times \mathrm{dom}(p'_{j_2}) \times \cdots \times \mathrm{dom}(p'_{j_m}), \tag{47}$$

gdzie:

- $\mathrm{dom}(p'_{j_m})$ to dziedzina parametrów wynikowych obiektu o_j,
- m to liczba parametrów, które są modyfikowane w przestrzeni stanu drugiego obiektu przez T_k.

3. **Równoważność na podstawie przestrzeni wejść i wyjść** Równoważność funkcji przejścia $\mathcal{T}$ i $\mathcal{T}'$ na podstawie przestrzeni wejść i wyjść jest zdefiniowana jako:

$$\mathcal{T} \equiv_{\mathrm{we/wy}} \mathcal{T}' \iff \forall T_k \in \mathcal{T}, \exists T'_k \in \mathcal{T}' : \begin{cases} \mathrm{dom}(T_k) = \mathrm{dom}(T'_k), \\ \mathrm{codom}(T_k) = \mathrm{codom}(T'_k). \end{cases} \tag{48}$$

Definicja ta oznacza, że przestrzenie wejść i wyjść funkcji przejścia T_k i T'_k są równoważne wtedy i tylko wtedy, gdy odpowiadające sobie parametry obiektów w procesach algebraicznych $\mathcal{P}$ i $\mathcal{P}'$ mają zgodne dziedziny:

$$\forall o_i \in \mathcal{O}, \exists o'_i \in \mathcal{O}' : \mathrm{dom}(p_j) = \mathrm{dom}(p'_j). \tag{49}$$

4. **Równoważność zachowania jako szczególny przypadek** Równoważność zachowania to szczególny przypadek równoważności przestrzeni wejść i wyjść, gdzie dodatkowo funkcje T_k i T'_k muszą zwracać identyczne wyniki dla tych samych parametrów wejściowych:

$$\mathcal{T} \equiv_{\mathrm{zachowanie}} \mathcal{T}' \iff \forall T_k \in \mathcal{T}, \exists T'_k \in \mathcal{T}' : \begin{cases} \mathrm{dom}(T_k) = \mathrm{dom}(T'_k), \\ \mathrm{codom}(T_k) = \mathrm{codom}(T'_k), \\ T_k(x) = T'_k(x), \quad \forall x \in \mathrm{dom}(T_k). \end{cases} \tag{50}$$

Dowód, że równoważność zachowania wynika z równoważności wejść i wyjść Załóżmy, że funkcje przejścia T_k i T'_k są równoważne w sensie zachowania:

$$\mathcal{T} \equiv_{\mathrm{zachowanie}} \mathcal{T}'. \tag{51}$$

Równoważność wyników $T_k(x) = T'_k(x)$ dla $x \in \mathrm{dom}(T_k)$ wymaga zgodności przestrzeni wejść $\mathrm{dom}(T_k) = \mathrm{dom}(T'_k)$ oraz przestrzeni wyjść $\mathrm{codom}(T_k) = \mathrm{codom}(T'_k)$. Stąd:

$$\mathcal{T} \equiv_{\mathrm{zachowanie}} \mathcal{T}' \implies \mathcal{T} \equiv_{\mathrm{we/wy}} \mathcal{T}'. \tag{52}$$

Kontrprzykład dla implikacji odwrotnej: Równoważność przestrzeni wejść i wyjść ($\mathcal{T} \equiv_{\mathrm{we/wy}} \mathcal{T}'$) nie implikuje równoważności zachowania, ponieważ $T_k(x) \neq T'_k(x)$ może zachodzić dla pewnych wartości $x \in \mathrm{dom}(T_k)$, mimo że przestrzenie wejść i wyjść są zgodne.

2.4 Równoważność Procesów Algebraicznych

Procesy algebraiczne $\mathcal{P}$ i $\mathcal{P}'$ są równoważne, jeśli:

$$\mathcal{P} \equiv \mathcal{P}' \iff (\mathcal{O} \equiv \mathcal{O}') \wedge (\mathcal{R} \equiv \mathcal{R}') \wedge (\mathcal{T} \equiv \mathcal{T}'). \tag{53}$$

3 Klasyfikacja zmian procesu

W celu dokładniejszego zbadania procesu algebraicznego można dokonać klasyfikacji możliwych modyfikacji, identyfikując różne typy zmian w jego podstawowych komponentach. Analiza ta pozwala określić, w jaki sposób poszczególne rodzaje zmian wpływają na strukturę procesu oraz jakie rodziny procesów powstają w wyniku tych modyfikacji. Dzięki temu możliwe jest lepsze zrozumienie zarówno lokalnych, jak i globalnych konsekwencji tych zmian. Niniejsza sekcja analizuje charakter tych modyfikacji oraz ich potencjalne konsekwencje dla własności, zachowania i struktury wynikowych rodzin procesów.

3.1 Parametry i Zmiany Parametryczne

Zrozumienie zmian parametrycznych w procesie algebraicznym $\mathcal{P} = (\mathcal{O}, \mathcal{R}, \mathcal{T})$ wymaga wcześniejszego poznania podziału parametrów w przestrzeniach stanów S_i. Parametry te opisują właściwości, dynamikę oraz bieżący stan obiektów $o_i \in \mathcal{O}$, stanowiąc podstawę analizy i modyfikacji procesu.

3.1.1 Klasyfikacja Parametrów w Przestrzeniach Stanów

Każdy obiekt o_i jest powiązany z przestrzenią stanów S_i, która może być podzielona na dwie główne kategorie parametrów: sterowalne i niesterowalne.

Parametry Sterowalne: Parametry sterowalne to takie, które można modyfikować w celu wpływania na przebieg procesu. Każdy parametr sterowalny jest funkcją czasu $p(t)$, gdzie:

- Jeśli $p(t) = \text{const}$, parametr jest statyczny.
- Jeśli $p(t)$ zmienia się w czasie, parametr jest dynamiczny.

$$S_{\text{sterowalne}} = \{p_1(t), p_2(t), \ldots, p_k(t)\} \subseteq S_i, \tag{54}$$

$$\text{dla wszystkich } p_j(t) \in S_{\text{sterowalne}}, \quad \exists t_1, t_2 : p_j(t_1) \neq p_j(t_2) \vee p_j(t_1) = p_j(t_2). \tag{55}$$

Przykłady:

- Maksymalna pojemność bufora: $p_{\text{bufor}}(t) = \text{const}$.
- Prędkość obrotowa wirnika: $p_{\text{predkosc}}(t) = f(t)$.
- Aktualne zużycie energii: $p_{\text{energia}}(t) = g(t)$.

Rola: Parametry sterowalne definiują trajektorię procesu i mogą być modyfikowane w celu jego optymalizacji lub dostosowania do zmieniających się warunków.

Parametry Niesterowalne: Parametry niesterowalne odzwierciedlają bieżący stan obiektu i nie podlegają bezpośredniemu sterowaniu. Są również funkcjami czasu $p(t)$, ale ich wartości zmieniają się wyłącznie w wyniku działania procesu.

$$S_{\text{niesterowalne}} = \{p_{k+1}(t), p_{k+2}(t), \ldots, p_m(t)\} \subseteq S_i, \tag{56}$$

Przykłady:

- Aktualne zapełnienie bufora: $p_{\text{zapełnienie\_bufora}}(t)$.
- Liczba wyprodukowanych części: $p_{\text{liczba\_części}}(t)$.
- Poziom energii w urządzeniu: $p_{\text{energia\_pozostała}}(t)$.

Rola: Parametry niesterowalne służą do monitorowania przebiegu procesu i umożliwiają analizę efektywności oraz jakości działania.

Reprezentacja Przestrzeni Stanów Pełna przestrzeń stanów obiektu o_i jest zbiorem wszystkich parametrów sterowalnych i niesterowalnych:

$$S_i = S_{\text{sterowalne}} \cup S_{\text{niesterowalne}}. \tag{57}$$

Ten podział pozwala na elastyczne modelowanie procesów algebraicznych, uwzględniając zarówno parametry kontrolujące przebieg procesu, jak i te służące do monitorowania bieżącego stanu.

3.1.2 Zmiany Parametryczne

Zmiany parametryczne odnoszą się do modyfikacji wartości parametrów stanu $p_j(t) \in s_i$ w obrębie obiektów $o_i \in \mathcal{O}$.

Definicja Niech $\mathcal{P}$ będzie procesem algebraicznym. Zmiana parametryczna zdefiniowana jest jako modyfikacja trajektorii czasowej parametru stanu $p_j(t) \in s_i$ obiektu $o_i \in \mathcal{O}$, co prowadzi do powstania procesu $\mathcal{P}'$:

$$\mathcal{P}' = (\mathcal{O}', \mathcal{R}, \mathcal{T}), \tag{58}$$

Dla każdego obiektu $o_i \in \mathcal{O}$, jego stan s_i w chwili t można zapisać jako zbiór parametrów:

$$s_i(t) = \{p_1(t), p_2(t), \ldots, p_n(t)\},$$

gdzie:

- $\{p_1(t), p_2(t), \ldots, p_k(t)\} \subseteq S_{\text{sterowalne}}$ — parametry sterowalne,
- $\{p_{k+1}(t), \ldots, p_n(t)\} \subseteq S_{\text{niesterowalne}}$ — parametry niesterowalne.

Zmiana parametryczna obejmuje modyfikację trajektorii czasowej parametrów $p_j(t)$, co może obejmować zarówno chwilowe zmiany wartości, jak i globalne modyfikacje trajektorii w czasie.

1. Zmiana wartości początkowej Modyfikacja wartości parametru w chwili początkowej $t = 0$:

$$p_j(0) \to p'_j(0), \quad \forall p_j \in s_i, \tag{59}$$

gdzie $p'_j(0)$ oznacza zaktualizowaną wartość początkową parametru p_j.

2. Zmiana wartości w określonej chwili t_0 Modyfikacja wartości parametru w dowolnym momencie t_0 bez wpływu na inne chwile:

$$p_j(t) = \begin{cases} p'_j(t), & t = t_0, \\ p_j(t), & t \neq t_0. \end{cases} \tag{60}$$

3. Zmiana trajektorii czasowej Globalna modyfikacja trajektorii czasowej parametru w czasie:

$$p_j(t) \to p'_j(t), \quad \forall t, \tag{61}$$

gdzie $p'_j(t)$ to zaktualizowana funkcja opisująca zależność parametru od czasu.

Przykłady Zmian Parametrycznych

- **Zmiana wartości początkowej:** Maksymalna pojemność bufora zmienia się z $p_{\text{bufor}}(0) = 100$ jednostek na $p'_{\text{bufor}}(0) = 150$ jednostek.
- **Zmiana wartości chwilowej:** Poziom energii w urządzeniu w chwili $t_0 = 10$ zmienia się z $p_{\text{energia}}(10) = 400$ na $p'_{\text{energia}}(10) = 450$, pozostając niezmienny w innych chwilach.
- **Zmiana trajektorii czasowej:** Prędkość obrotowa wirnika zmienia swoją zależność czasową z $p_{\text{predkosc}}(t) = 3000 \cdot \sin(t)$ na $p'_{\text{predkosc}}(t) = 4000 \cdot \cos(t)$.

Każdy z powyższych przypadków może być interpretowany jako zmiana trajektorii czasowej parametru. Zmiana wartości początkowej lub chwilowej są szczególnymi przypadkami, w których trajektoria modyfikowana jest tylko w określonych punktach czasu.

3.1.3 Rodzina parametryczna procesu $\mathcal{P}$

Zmiana parametryczna prowadzi do modyfikacji wartości parametrów $p_j(t)$ w czasie, co powoduje, że stany obiektów s_i różnią się w procesach $\mathcal{P}$ i $\mathcal{P}'$. Zgodnie z kryterium równoważności wartości parametrów (28), które wymaga, aby odpowiadające sobie obiekty $o_i \in \mathcal{O}$ i $o'_i \in \mathcal{O}'$ miały identyczne wartości parametrów w swoich stanach, zmiana parametryczna prowadzi do jego złamania. Formalnie:

$$\exists t \geq 0,\ p_j(t) \neq p'_j(t) \implies s_i(t) \neq s'_i(t). \tag{62}$$

W efekcie obiekty $o_i \in \mathcal{O}$ i $o'_i \in \mathcal{O}'$ nie są równoważne w sensie wartości parametrów:

$$\mathcal{O} \not\equiv_{\text{parametry}} \mathcal{O}'. \tag{63}$$

Wprowadzenie zmiany parametrycznej w procesie algebraicznym $\mathcal{P}$ prowadzi do powstania nowego procesu $\mathcal{P}'$, który różni się od $\mathcal{P}$ w sensie ustalonych wartości parametrów. Zbiór wszystkich procesów powstałych w wyniku zmian parametrycznych nazywamy rodziną parametryczną procesu $\mathcal{P}$.

Definicja rodziny parametrycznej Niech $\mathcal{P}$ będzie procesem algebraicznym. Rodzina parametryczna $\mathcal{F}_{\text{param}}(\mathcal{P})$ jest zbiorem procesów $\mathcal{P}'$, które różnią się od $\mathcal{P}$ w sensie wartości parametrów, co oznacza złamanie kryterium równoważności wartości parametrów.

Formalnie, rodzina parametryczna procesu $\mathcal{P}$ jest zdefiniowana jako:

$$\mathcal{F}_{\text{param}}(\mathcal{P}) = \{\mathcal{P}' = (\mathcal{O}', \mathcal{R}, \mathcal{T}) \mid \mathcal{O} \not\equiv_{\text{parametry}} \mathcal{O}'\}. \tag{64}$$

Wnioski Rodzina parametryczna $\mathcal{F}_{\text{param}}(\mathcal{P})$ tworzy klasę abstrakcji procesów algebraicznych, w której jedynym złamanym kryterium równoważności jest kryterium wartości parametrów. Procesy należące do tej klasy są odrębne pod względem konfiguracji parametrów w czasie, lecz równoważne w sensie przestrzeni stanów i struktury. Klasa abstrakcji pozwala na grupowanie procesów, które różnią się trajektoriami parametrów, lecz zachowują wspólne właściwości strukturalne.

Równoważność rodzin parametrycznych Rodziny parametryczne $\mathcal{F}_{\text{param}}(\mathcal{P})$ i $\mathcal{F}_{\text{param}}(\mathcal{P}')$ są równoważne, jeśli wszystkie procesy wchodzące w skład tych rodzin zachowują kryterium dziedzin parametrów. Formalnie, równoważność rodzin parametrycznych definiujemy jako:

$$\mathcal{F}_{\text{param}}(\mathcal{P}) \equiv \mathcal{F}_{\text{param}}(\mathcal{P}') \iff \forall \mathcal{P} \in \mathcal{F}_{\text{param}}(\mathcal{P}),\ \forall \mathcal{P}' \in \mathcal{F}_{\text{param}}(\mathcal{P}') : \mathcal{O} \equiv_{\text{dziedzina}} \mathcal{O}'. \tag{65}$$

W praktyce oznacza to, że procesy wchodzące w skład równoważnych rodzin parametrycznych mają identyczne dziedziny parametrów w przestrzeniach stanów. Równoważność rodzin parametrycznych zachodzi, gdy dziedziny parametrów pozostają niezmienione, niezależnie od różnic w wartościach parametrów.

Złamanie kryterium dziedzin parametrów Jeśli kryterium dziedzin parametrów zostanie złamane, to istnieje przynajmniej jeden proces $\mathcal{P}' \in \mathcal{F}_{\text{param}}(\mathcal{P}')$, którego dziedziny parametrów różnią się od dziedzin parametrów procesów w $\mathcal{F}_{\text{param}}(\mathcal{P})$. Formalnie:

$$\exists \mathcal{P} \in \mathcal{F}_{\text{param}}(\mathcal{P}),\ \exists \mathcal{P}' \in \mathcal{F}_{\text{param}}(\mathcal{P}') : \mathcal{O} \not\equiv_{\text{dziedzina}} \mathcal{O}'. \tag{66}$$

W takim przypadku rodziny $\mathcal{F}_{\text{param}}(\mathcal{P})$ i $\mathcal{F}_{\text{param}}(\mathcal{P}')$ nie są równoważne:

$$\mathcal{F}_{\text{param}}(\mathcal{P}) \not\equiv \mathcal{F}_{\text{param}}(\mathcal{P}'). \tag{67}$$

Wnioski Równoważność rodzin parametrycznych opiera się na zachowaniu kryterium dziedzin parametrów. Gdy to kryterium zostaje złamane, powstaje nowa rodzina parametryczna $\mathcal{F}'_{\text{param}}$, która nie jest równoważna z pierwotną rodziną $\mathcal{F}_{\text{param}}$. Zmiana dziedzin parametrów wpływa na strukturę przestrzeni stanów, co redefiniuje zbiór możliwych procesów i prowadzi do powstania nowej klasy abstrakcji.

Wpływ zmiany dziedzin parametrów na funkcje przejścia Zmiana dziedzin parametrów w przestrzeniach stanów procesu algebraicznego $\mathcal{P}$ prowadzi nie tylko do redefinicji przestrzeni stanów, ale także do modyfikacji zbioru funkcji przejścia $\mathcal{T}$. Funkcje przejścia $T_k \in \mathcal{T}$ są zdefiniowane na podstawie dziedzin parametrów w przestrzeniach stanów obiektów, które łączą. Formalnie, dziedzina funkcji przejścia T_k wynika z przestrzeni wejściowej parametrów:

$$\operatorname{dom}(T_k) = \prod_{p_j \in S_i} \operatorname{dom}(p_j),$$

gdzie S_i jest przestrzenią stanów obiektu o_i, a p_j to parametr tej przestrzeni.

Zmiana dziedziny parametru p_j z $\operatorname{dom}(p_j)$ na $\operatorname{dom}'(p_j)$ prowadzi do zmiany dziedziny funkcji przejścia T_k:

$$\operatorname{dom}(T_k) \neq \operatorname{dom}'(T_k).$$

Analogicznie, przeciwdziedzina $\operatorname{codom}(T_k)$, która opisuje przestrzeń wynikową, również może zostać zmodyfikowana w wyniku zmiany dziedzin parametrów wynikowych:

$$\operatorname{codom}(T_k) \neq \operatorname{codom}'(T_k).$$

Złamanie równoważności funkcji przejścia Jeśli zmiana dziedzin parametrów prowadzi do różnic w przestrzeniach wejściowych i/lub wynikowych funkcji przejścia, zostaje złamane kryterium równoważności funkcji przejścia w sensie przestrzeni wejść i wyjść:

$$\mathcal{T} \not\equiv_{\text{we/wy}} \mathcal{T}'.$$

Formalnie, kryterium równoważności przestrzeni wejść i wyjść funkcji przejścia nie jest spełnione, jeśli:

$$\exists T_k \in \mathcal{T},\ \exists T'_k \in \mathcal{T}' : \operatorname{dom}(T_k) \neq \operatorname{dom}(T'_k) \lor \operatorname{codom}(T_k) \neq \operatorname{codom}(T'_k).$$

Konsekwencje dla rodzin parametrycznych Zmiana dziedzin parametrów w przestrzeniach stanów prowadzi do powstania nowej rodziny parametrycznej $\mathcal{F}'_{\text{param}}$, która nie jest równoważna z pierwotną rodziną $\mathcal{F}_{\text{param}}$. Ponadto, nowa rodzina parametryczna generuje zbiór funkcji przejścia $\mathcal{T}'$, który różni się od pierwotnego zbioru $\mathcal{T}$ pod względem przestrzeni wejść i wyjść. Formalnie:

$$\mathcal{F}_{\text{param}}(\mathcal{P}) \not\equiv \mathcal{F}'_{\text{param}}(\mathcal{P}) \implies \mathcal{T} \not\equiv_{\text{we/wy}} \mathcal{T}'.$$

Wnioski Zmiana dziedzin parametrów ma istotny wpływ na strukturę procesu algebraicznego. Naruszenie równoważności przestrzeni wejść i wyjść funkcji przejścia prowadzi do powstania nowych trajektorii procesu, redefiniując zarówno przestrzenie stanów, jak i funkcje przejścia. W efekcie tworzy to nową klasę abstrakcji procesów algebraicznych, która obejmuje zarówno zmodyfikowane przestrzenie stanów, jak i zmodyfikowane funkcje przejścia.

—

4 Atlas Procesu Algebraicznego

Każdy obiekt $o_i \in \mathcal{O}$ w procesie algebraicznym charakteryzuje się zestawem parametrów, które zmieniają się w czasie. Przestrzeń stanów S_i tego obiektu to zbiór możliwych **wektorów stanu** $\mathbf{s}_i(t)$, które opisują ewolucję jego parametrów w czasie.

Dynamika obiektu wynika z sekwencji wektorów stanu przyjmowanych w kolejnych chwilach t, co prowadzi do kluczowych pojęć:

- **Mapa stanów** – zbiór wszystkich wektorów stanu $\mathbf{s}_i(t)$, określający pełny opis ewolucji obiektu.
- **Trajektoria stanu** – zbiór wartości pojedynczego parametru $p_{ij}(t)$ w czasie.

Atlas procesu algebraicznego obejmuje zbiór map stanów wszystkich obiektów o_i w systemie, stanowiąc pełną reprezentację dynamiki procesu oraz zależności między jego elementami w czasie.

4.1 Wektory Stanu

Definicja: Wektor stanu $\mathbf{s}_i(t)$ obiektu o_i w chwili t to wektor wartości parametrów tego obiektu:

$$\mathbf{s}_i(t) = [p_{i1}(t), p_{i2}(t), \ldots, p_{in}(t)]^\top \in \mathbb{R}^n. \tag{68}$$

Początkowy stan obiektu jest dany przez:

$$\mathbf{s}_i(0) = [p_{i1}(0), p_{i2}(0), \ldots, p_{in}(0)]^\top. \tag{69}$$

4.2 Trajektoria Stanu

Definicja: Trajektoria stanu to ewolucja wartości pojedynczego parametru $p_{ij}(t)$ w czasie:

$$\mathcal{T}_{ij} = \{p_{ij}(t) \mid t \geq 0\}. \tag{70}$$

Każda trajektoria reprezentuje sposób, w jaki dany parametr zmienia się w czasie, a jej kształt zależy od:

- **Warunków początkowych** – wartości początkowej parametru $p_{ij}(0)$.
- **Natury parametru** – czy jest sterowalny, niesterowalny, stały lub zmienny.
- **Funkcji przejścia** – opisującej zmiany stanu w czasie.

4.3 Mapa Stanów

Definicja: Mapa stanów M_i obiektu o_i może być definiowana na dwa sposoby:

1. **Jako zbiór wektorów stanu**, opisujących stan obiektu w każdej chwili t:

$$M_i = \{\mathbf{s}_i(t) \mid t \geq 0\}. \tag{71}$$

2. **Jako zbiór trajektorii** wszystkich parametrów obiektu w czasie:

$$M_i = \{\mathcal{T}_{ij} \mid j = 1, \ldots, n\}. \tag{72}$$

Reprezentacja macierzowa mapy stanów Mapa stanów może być przedstawiona w postaci macierzy, w której wiersze lub kolumny mogą reprezentować różne podejścia:

- **Podejście kolumnowe** (trajektorie parametrów): Każda kolumna odpowiada trajektorii jednego parametru $p_{ij}(t)$, co oznacza, że wiersze reprezentują wartości parametrów w danej chwili czasu t_k:

$$M_i = \begin{bmatrix} p_{i1}(t_1) & p_{i2}(t_1) & \ldots & p_{in}(t_1) \\ p_{i1}(t_2) & p_{i2}(t_2) & \ldots & p_{in}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_{i1}(t_m) & p_{i2}(t_m) & \ldots & p_{in}(t_m) \end{bmatrix}. \tag{73}$$

- **Podejście wierszowe** (wektory stanu): Każdy wiersz macierzy odpowiada wektorowi stanu $\mathbf{s}_i(t_k)$ w konkretnej chwili czasu t_k, czyli zawiera wartości wszystkich parametrów dla danego momentu:

$$M_i = \begin{bmatrix} \mathbf{s}_i(t_1)^\top \\ \mathbf{s}_i(t_2)^\top \\ \vdots \\ \mathbf{s}_i(t_m)^\top \end{bmatrix}. \tag{74}$$

Relacja między pojęciami

- **Wektor stanu** $\mathbf{s}_i(t)$ opisuje wartości wszystkich parametrów obiektu w danej chwili t.
- **Trajektoria stanu** $\mathcal{T}_{ij}$ to ewolucja jednego parametru $p_{ij}(t)$ w czasie.
- **Mapa stanów** M_i to zbiór wszystkich wektorów stanu (podejście wierszowe) lub zbiór trajektorii parametrów (podejście kolumnowe).

4.4 Ewolucja Wektorów Stanu

W procesach algebraicznych wektor stanu $\mathbf{s}_i(t)$ może ewoluować na dwa sposoby:

1. **Z góry ustalone trajektorie** – parametry ewoluują niezależnie, zgodnie z określoną funkcją czasu.
2. **Zmiany wynikające z funkcji przejścia** – parametry zmieniają się w wyniku interakcji z innymi obiektami.

Ewolucja zgodna z trajektorią Dla pewnych parametrów zmiany są znane z góry i nie wynikają z interakcji w procesie. Możemy to zapisać jako:

$$p_{ij}(t+1) = f_{ij}(t), \tag{75}$$

gdzie $f_{ij}(t)$ to funkcja opisująca zmianę parametru p_{ij} w czasie.

Przykłady:

- **Czas systemowy** – zegar procesu może zmieniać się liniowo $t \to t+1$.
- **Temperatura otoczenia** – może zmieniać się zgodnie z ustalonym harmonogramem.
- **Czas cyklu maszyny** – może być stały

Ewolucja wynikająca z funkcji przejścia Parametry mogą również zmieniać się w wyniku interakcji z innymi obiektami procesu. Jest to opisane funkcją przejścia T_i:

$$\mathbf{s}_i(t+1) = T_i(\mathbf{s}_i^{(I)}(t), \mathbf{u}_i(t)), \tag{76}$$

gdzie:

- $\mathbf{s}_i^{(I)}(t)$ oznacza podzbiór parametrów wektora stanu, które wpływają na jego ewolucję,
- $I \subseteq \{1, 2, \ldots, n\}$ to indeksy parametrów wykorzystywanych w funkcji przejścia,
- $\mathbf{u}_i(t)$ to zewnętrzne wpływy na obiekt.

Przykłady:

- **Obciążenie maszyny** – może zależeć od ilości obrabianych części i wpływu innych operacji.
- **Stan magazynu** – zależy od wejścia surowców i ich zużycia w produkcji.
- **Poziom zużycia narzędzia** – zmienia się w wyniku obróbki, ale jego zużycie może zależeć od wielu czynników, np. materiału obrabianego.

Podział parametrów ze względu na sposób ewolucji Każdy parametr $p_{ij}(t)$ można zaklasyfikować do jednej z trzech grup:

- **Parametry statyczne** – nie zmieniają się w czasie, np. stałe systemowe.
- **Parametry trajektoryjne** – zmieniają się zgodnie z ustaloną funkcją czasu $f_{ij}(t)$.
- **Parametry dynamiczne** – zmieniają się w wyniku funkcji przejścia T_i.

Reprezentacja macierzowa ewolucji wektorów stanu Jeśli rozważymy zbiór N obiektów w procesie, możemy opisać ich ewolucję w postaci macierzowej:

$$S(t+1) = \begin{bmatrix} T_1(S_1^{(I)}(t), U_1(t)) \\ T_2(S_2^{(I)}(t), U_2(t)) \\ \vdots \\ T_N(S_N^{(I)}(t), U_N(t)) \end{bmatrix} + \begin{bmatrix} f_{11}(t) \\ f_{21}(t) \\ \vdots \\ f_{N1}(t) \end{bmatrix}, \tag{77}$$

gdzie:

- Pierwszy składnik to zmiany wynikające z funkcji przejścia,
- Drugi składnik to zmiany wynikające z deterministycznych trajektorii.

Przykład: Proces produkcji z różnymi typami parametrów Rozważmy system produkcyjny składający się z:

- **Maszyny** (o_1), której temperatura (p_{11}) zmienia się w sposób naturalny (*trajektoria*),
- **Magazynu** (o_2), którego poziom zapasów (p_{21}) zmienia się w wyniku operacji (*funkcja przejścia*),
- **Pracownika** (o_3), którego dostępność (p_{31}) jest stała (*parametr statyczny*).

Ewolucja stanu systemu można przedstawić jako:

$$\begin{bmatrix} p_{11}(t+1) \\ p_{21}(t+1) \\ p_{31}(t+1) \end{bmatrix} = \begin{bmatrix} f_{11}(t) \\ T_2(p_{21}(t), \mathbf{u}_2(t)) \\ p_{31}(t) \end{bmatrix}. \tag{78}$$

Interpretacja:

- $p_{11}(t)$ (temperatura maszyny) zmienia się zgodnie z trajektorią $f_{11}(t)$.
- $p_{21}(t)$ (poziom zapasów) zależy od operacji produkcyjnych i ewoluuje na podstawie funkcji przejścia.
- $p_{31}(t)$ (dostępność pracownika) pozostaje stała.

4.5 Atlas Procesu Algebraicznego

Atlas Procesu Algebraicznego jest zdefiniowany jako zbiór map stanów wszystkich obiektów procesu:

$$\mathcal{A}_P = \{M_i \mid o_i \in \mathcal{O}\}, \tag{79}$$

gdzie każda mapa M_i opisuje trajektorie stanu indywidualnego obiektu w czasie.

Atlas reprezentuje pełen zbiór trajektorii systemu w jego przestrzeni stanów. Każda mapa M_i odzwierciedla dynamikę i ewolucję parametrów obiektu o_i, a zbiór tych map umożliwia analizę globalnych właściwości systemu.

Atlas dla Skończonych Przestrzeni Stanów W rzeczywistych zastosowaniach przestrzenie stanów S_i obiektów $o_i \in \mathcal{O}$ są ograniczone, co wynika z dyskretnych kroków czasowych, skończonej liczby parametrów opisujących obiekty oraz skończoności dziedzin tych parametrów. Oznacza to, że każda wartość parametru $p_{ij}(t)$ przyjmuje wartości z ograniczonego zbioru:

$$p_{ij}(t) \in D_{ij}, \quad \text{gdzie } D_{ij} \text{ jest skończonym podzbiorem } \mathbb{R}. \tag{80}$$

Dla każdego obiektu o_i, mapa stanów M_i przyjmuje postać:

$$M_i = \{\mathbf{s}_i(t_k) \mid t_k \in T,\, k = 1, \ldots, N\}, \tag{81}$$

gdzie:

- $T = \{t_1, t_2, \ldots, t_N\}$ jest skończonym zbiorem kroków czasowych,
- N oznacza liczbę kroków czasowych w symulacji,
- $\mathbf{s}_i(t_k) \in D_{i1} \times D_{i2} \times \ldots \times D_{in}$, co oznacza, że każdy parametr ma ograniczoną liczbę możliwych wartości.

Zastosowanie Atlasu dla Skończonych Przestrzeni Stanów Atlas dla skończonych przestrzeni stanów znajduje zastosowanie w praktycznych symulacjach procesów produkcyjnych, gdzie:

- Analizowane są trajektorie parametrów w określonym przedziale czasowym,
- Oceniane są właściwości dynamiczne procesu, takie jak wąskie gardła, opóźnienia i zużycie zasobów,
- Uwzględniane są ograniczenia wartości parametrów wynikające z rzeczywistych możliwości systemu.

Przykład Rozważmy proces produkcyjny obejmujący maszynę, magazyn i moduł kontroli jakości:

- Maszyna: czas cyklu $p_{11}(t)$, czas przestoju $p_{12}(t)$,
- Magazyn: poziom zapasów $p_{21}(t)$,
- Kontrola jakości: liczba odrzuconych części $p_{31}(t)$.

Każdy parametr przyjmuje wartości z ograniczonej dziedziny:

$$p_{11}(t) \in D_{11}, \quad p_{21}(t) \in D_{21}, \quad p_{31}(t) \in D_{31}. \tag{82}$$

Przestrzeń stanów każdego obiektu jest skończona, a atlas opisuje ewolucję parametrów w dyskretnym czasie, uwzględniając ograniczenia wynikające z dziedzin parametrów, co umożliwia analizę i optymalizację całego procesu.

4.6 Mapowanie strumienia wartości

Bezpośrednio z definicji wektorów stanu wynika koncepcja *mapowania strumienia wartości*. Mapa strumienia wartości (VSM) dla procesu algebraicznego to zbiór wektorów stanu wszystkich obiektów w wybranej chwili t:

$$\mathcal{V}(t) = \{\mathbf{s}_1(t), \mathbf{s}_2(t), \ldots, \mathbf{s}_N(t)\}. \tag{83}$$

Związek z atlasem procesu algebraicznego: Atlas procesu algebraicznego $\mathcal{A}_P$ zawiera pełen zbiór map stanów wszystkich obiektów procesu, opisując ich trajektorie w czasie:

$$\mathcal{A}_P = \{M_i \mid o_i \in \mathcal{O}\}. \tag{84}$$

Mapa strumienia wartości (VSM) w istocie odpowiada wartości atlasu procesu algebraicznego w wybranym czasie t:

$$\mathcal{V}(t) = \mathcal{A}_P(t). \tag{85}$$

Ważne obserwacje:

- Mapa strumienia wartości (VSM) jest wartością atlasu procesu algebraicznego w chwili t.
- VSM jest szczególnym przypadkiem atlasu, który reprezentuje wycinek globalnej dynamiki systemu w danym momencie.
- Tworzenie VSM w praktyce oznacza zebranie wartości wektorów stanu w określonym czasie i ich zapis na mapie.

4.7 Analiza Wrażliwości w Procesie Algebraicznym

Definicja: Analiza wrażliwości w kontekście atlasu procesu algebraicznego polega na badaniu wpływu zmiany wartości wybranego parametru p_{ij} na ewolucję stanów obiektów i całego systemu. Badanie to opiera się na porównaniu atlasów procesu dla różnych wartości tego parametru.

Zbiór atlasów dla różnych wartości parametru Dla wybranego parametru p_{ij}, który podlega zmianom, definiujemy rodzinę atlasów:

$$\mathcal{A}_P^{(k)} = \{M_i^{(k)} \mid o_i \in \mathcal{O}\}, \quad k = 1, 2, \ldots, N, \tag{86}$$

gdzie $\mathcal{A}_P^{(k)}$ to atlas wygenerowany dla konkretnej wartości parametru $p_{ij}^{(k)}$. Zmienność parametru może być ustalana na podstawie przedziału wartości:

$$p_{ij}^{(k)} = p_{ij}^{(0)} + k \cdot \Delta p_{ij}, \quad k = 1, 2, \ldots, N. \tag{87}$$

Porównanie atlasów Porównanie atlasów $\mathcal{A}_P^{(k)}$ pozwala na określenie wpływu zmienności parametru na ewolucję stanów systemu. Aby ilościowo ocenić wrażliwość procesu na dany parametr, można zdefiniować miarę odchylenia między atlasami:

$$D(k) = \sum_i d(M_i^{(k)}, M_i^{(0)}), \tag{88}$$

gdzie $d(M_i^{(k)}, M_i^{(0)})$ to metryka odległości pomiędzy mapami stanów dla różnych wartości parametru, np. norma Frobeniusa:

$$d(M_i^{(k)}, M_i^{(0)}) = \|M_i^{(k)} - M_i^{(0)}\|_F = \sqrt{\sum_{t=1}^{T} \sum_{j=1}^{n} \left(p_{ij}^{(k)}(t) - p_{ij}^{(0)}(t)\right)^2}. \tag{89}$$

Interpretacja wyników Norma Frobeniusa pozwala określić **globalną różnicę** między atlasami procesu algebraicznego, dając ogólną miarę wpływu zmiany parametru p_{ij} na cały system. Jednak ta miara nie dostarcza informacji o tym, które konkretne parametry uległy największym zmianom. W celu uzyskania bardziej szczegółowego wglądu konieczne jest przeprowadzenie analizy wpływu poszczególnych parametrów, co prowadzi do rozkładu normy Frobeniusa.

4.7.1 Rozkład Normy Frobeniusa

Aby określić, które parametry mają największy wpływ na różnice między atlasami procesu algebraicznego, możemy rozłożyć normę Frobeniusa na wkłady poszczególnych parametrów.

Rozkład normy Frobeniusa na poziomie parametrów Dla macierzy stanów $M_i^{(k)}$ oraz $M_i^{(0)}$ różnica opisana normą Frobeniusa wynosi:

$$\|M_i^{(k)} - M_i^{(0)}\|_F = \sqrt{\sum_{j=1}^{n} \sum_{t=1}^{T} \left(p_{ij}^{(k)}(t) - p_{ij}^{(0)}(t)\right)^2}. \tag{90}$$

Aby określić, które parametry p_{ij} mają największy wpływ, możemy rozłożyć normę na wkłady każdego parametru:

$$d_j(M_i^{(k)}, M_i^{(0)}) = \sqrt{\sum_{t=1}^{T} \left(p_{ij}^{(k)}(t) - p_{ij}^{(0)}(t)\right)^2}. \tag{91}$$

Całkowita różnica między atlasami może być więc zapisana jako suma wkładów poszczególnych parametrów:

$$\|M_i^{(k)} - M_i^{(0)}\|_F = \sum_{j=1}^{n} d_j(M_i^{(k)}, M_i^{(0)}). \tag{92}$$

Normalizacja i ranking wpływu parametrów Aby porównać wpływ różnych parametrów, warto zastosować normalizację wkładów:

$$\tilde{d}_j = \frac{d_j}{\sum_{j=1}^{n} d_j} \times 100\%. \tag{93}$$

Dzięki temu możemy określić procentowy udział każdego parametru w całkowitej różnicy między atlasami, co pozwala na identyfikację kluczowych zmiennych wrażliwych na zmiany w systemie.

Interpretacja wyników - Jeśli $\tilde{d}_j$ dla danego parametru jest duże, oznacza to, że jego zmiana ma istotny wpływ na ewolucję systemu. - Jeśli $\tilde{d}_j \approx 0$ dla danego parametru, oznacza to, że jego zmiana nie wpływa istotnie na system. - Ranking parametrów pozwala na priorytetyzację działań optymalizacyjnych i regulacyjnych w analizowanym procesie.

Wykorzystanie analizy czasowej Zamiast sumować różnice po całym czasie, możemy analizować zmiany w danej chwili t:

$$D(t) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij}^{(k)}(t) - p_{ij}^{(0)}(t) \right)^2. \tag{94}$$

Dzięki temu możemy określić, w których momentach czasowych zmiany w systemie są największe.

4.7.2 Zagadnienia

Wpływ zmiany parametru na wartości końcowe Jednym z kluczowych aspektów analizy wrażliwości jest określenie, w jaki sposób zmiana wybranego parametru p_{ij} wpływa na końcowy stan procesu w zadanym czasie T_f. Porównujemy końcowe wartości wektorów stanu dla różnych wariantów wartości parametru:

$$\mathbf{s}_i^{(k)}(T_f) = \left[p_{i1}^{(k)}(T_f), p_{i2}^{(k)}(T_f), \ldots, p_{in}^{(k)}(T_f) \right]^\top. \tag{95}$$

Różnica między wartościami końcowymi dla różnych wariantów wartości parametru może być mierzona jako:

$$\Delta \mathbf{s}_i(T_f) = \mathbf{s}_i^{(k)}(T_f) - \mathbf{s}_i^{(0)}(T_f). \tag{96}$$

Dla poszczególnych parametrów możemy wyznaczyć odchylenie końcowe:

$$\Delta p_{ij}(T_f) = p_{ij}^{(k)}(T_f) - p_{ij}^{(0)}(T_f). \tag{97}$$

Jeśli różnice $\Delta p_{ij}(T_f)$ są duże, oznacza to, że parametr p_{ij} ma istotny wpływ na wartości końcowe procesu.

Czas Osiągnięcia Stanu Docelowego

Określenie czasu osiągnięcia oczekiwanej wartości Innym ważnym zagadnieniem jest określenie czasu T_{ij}^*, po jakim dany parametr osiągnie oczekiwaną wartość p_{ij}^*. Możemy zdefiniować ten czas jako:

$$T_{ij}^* = \min\{t \mid p_{ij}(t) = p_{ij}^*\}. \tag{98}$$

Jeśli parametr nigdy nie osiąga wartości p_{ij}^*, możemy wyznaczyć asymptotyczną wartość końcową:

$$p_{ij}^{\infty} = \lim_{t \to \infty} p_{ij}(t). \tag{99}$$

Analiza czasu osiągnięcia wartości pozwala ocenić, jak szybko system reaguje na zmiany parametrów oraz czy w ogóle możliwe jest osiągnięcie oczekiwanego stanu.

Interpretacja wyników - Jeśli $\Delta p_{ij}(T_f)$ jest duże, oznacza to, że parametr ma kluczowy wpływ na końcowy stan systemu. - Jeśli T^*_{ij} jest krótki, oznacza to, że system szybko reaguje na zmiany danego parametru. - Jeśli parametr nigdy nie osiąga wartości p^*_{ij}, może to wskazywać na istnienie ograniczeń w dynamice procesu.

5 Uogólnienia procesu algebraicznego - Czynność

5.1 Motywacja dla pojęcia czynności

Cel wprowadzenia pojęcia czynności W klasycznym ujęciu procesu algebraicznego obiekty o_i są traktowane jako jednostki o określonych stanach i ewoluujących parametrach. Jednak w wielu zastosowaniach rzeczywistych, szczególnie w procesach biznesowych i biurowych, trudniej jest jednoznacznie wskazać fizyczne obiekty systemu. W takich przypadkach użycie pojęcia **czynności** pozwala na alternatywną reprezentację struktury procesu.

Czynność jako abstrakcyjny obiekt W kontekście procesów biznesowych, czynność może być rozumiana jako element strukturalny systemu, który:

- Nie posiada jednoznacznej reprezentacji fizycznej (np. „zatwierdzenie faktury", „analiza danych"),
- Jest powiązany z określoną transformacją informacji lub zasobów,
- Tworzy sekwencję działań prowadzących do określonego celu w procesie.

Proces algebraiczny a czynność Pojęcie czynności stanowi alternatywny sposób myślenia o procesie algebraicznym. Zamiast analizować ewolucję stanów obiektów, można traktować czynności jako główne jednostki strukturalne procesu:

$$P = \{C_1, C_2, \ldots, C_n\}, \tag{100}$$

gdzie P jest procesem, a C_i to czynności składające się na jego przebieg. Taka reprezentacja jest szczególnie użyteczna w sytuacjach, gdzie obiekty procesu są abstrakcyjne, a same czynności są kluczowe dla opisu jego struktury.

Intuicyjność pojęcia czynności Jednym z głównych powodów wprowadzenia czynności jako elementu opisu procesu algebraicznego jest ułatwienie analizy procesów abstrakcyjnych, takich jak:

- Przepływy dokumentów w firmie,
- Decyzje podejmowane przez ludzi lub systemy AI,
- Optymalizacja procesów administracyjnych,
- Automatyzacja w systemach workflow.

Dzięki temu podejście oparte na czynnościach pozwala na bardziej intuicyjne zrozumienie złożonych procesów, w których nie występują klasyczne, fizyczne obiekty.

Przykłady zastosowań Przykłady procesów, w których reprezentacja za pomocą czynności jest bardziej naturalna niż klasyczne podejście do procesów algebraicznych:

1. **Przetwarzanie zamówienia:** $C_1 =$ „Wprowadzenie zamówienia"
   $C_2 =$ „Weryfikacja dostępności produktu"
   $C_3 =$ „Zatwierdzenie zamówienia"
   $C_4 =$ „Przekazanie do realizacji"

2. **Analiza danych:** C_1 = „Pobranie danych”
   C_2 = „Wstępne przetworzenie”
   C_3 = „Analiza statystyczna”
   C_4 = „Wygenerowanie raportu”

Podsumowanie Wprowadzenie pojęcia czynności jako obiektu w procesie algebraicznym pozwala na bardziej abstrakcyjne spojrzenie na jego strukturę. Dzięki temu możliwe jest intuicyjne modelowanie procesów, które nie posiadają jednoznacznej reprezentacji fizycznej, ale są kluczowe dla zrozumienia dynamiki systemu.

5.2 Ewolucja procesów: Realne, Teoretyczne i Emergentne

Motywacja: Procesy w organizacji nie są statyczne – podlegają ciągłej optymalizacji i ewolucji. Każda optymalizacja nie tylko redukuje czas trwania czynności, ale może również prowadzić do pojawienia się nowych procesów, które wcześniej były niemożliwe do wykonania lub nawet nie były brane pod uwagę.

Wyróżniamy trzy poziomy procesów:

- **Procesy Realne** – istniejące i wykonywane procesy, które można zoptymalizować.
- **Procesy Teoretyczne** – procesy, które były technicznie możliwe, ale wcześniej były nieopłacalne lub zbyt czasochłonne.
- **Procesy Emergentne** – procesy, które pojawiły się w wyniku wdrożenia nowej technologii i nie były wcześniej nawet rozważane.

5.2.1 1 Procesy Realne – optymalizacja istniejących działań

Definicja: Są to procesy, które już istnieją w firmie i są aktywnie wykonywane. Ich struktura jest znana, a optymalizacja polega na eliminacji marnotrawstwa, poprawie efektywności i redukcji czasu trwania.

Przykład:

- Obliczenia wykonywane ręcznie, zajmujące 5h miesięcznie.
- Pracownik musiał manualnie analizować dane, co pochłaniało czas i było podatne na błędy.
- Po wdrożeniu aplikacji czas operacji został skrócony do kilku minut.

Metody optymalizacji:

- Standaryzacja działań.
- Automatyzacja powtarzalnych czynności.
- Eliminacja zbędnych kroków procesu.

5.2.2 2 Procesy Teoretyczne – możliwe, ale wcześniej nieopłacalne

Definicja: To procesy, które były **teoretycznie możliwe do wdrożenia**, ale ich realizacja była zbyt czasochłonna, kosztowna lub wymagała zbyt wielu zasobów. Dopiero wdrożenie optymalizacji sprawia, że stają się one wykonalne i opłacalne.

Przykład:

- Wcześniej można było ręcznie analizować dane, ale wymagałoby to żmudnych obliczeń, więc w praktyce nikt tego nie robił.
- Po wdrożeniu aplikacji system natychmiast wyznacza kluczowe wskaźniki w czasie rzeczywistym.
- Proces, który wcześniej był teoretycznie możliwy, teraz staje się rzeczywistością.

Metody wdrożenia:

- Automatyzacja poprzez algorytmy i AI.
- Wykorzystanie systemów predykcyjnych.
- Integracja istniejących baz danych i automatyzacja przepływu informacji.

5.2.3 3 Procesy Emergentne – całkowicie nowe procesy, które nie istniały wcześniej

Definicja: To procesy, które wcześniej były **nawet niebrane pod uwagę**, ponieważ brakowało ku temu narzędzi, danych lub technologii. Dopiero wdrożenie optymalizacji otworzyło nowe możliwości.

Przykład:

- Po wdrożeniu aplikacji użytkownicy mogą analizować dane w czasie rzeczywistym.
- Pojawiają się **nowe metody pracy**, których wcześniej nikt nie rozważał.
- Powstają **nowe pomysły** na ulepszenie systemu, które wcześniej były nieznane.

Metody wdrożenia:

- Zaawansowana analityka predykcyjna.
- Uczenie maszynowe i analiza big data.
- Internet rzeczy (IoT) i inteligentne systemy raportowania.

5.3 Warstwowy rozwój procesu

Każdy proces może ewoluować, przechodząc przez kolejne poziomy:

Rodzaj procesu	Stan przed wdrożeniem	Stan po wdrożeniu
Realny	Proces istniejący, wykonywany ręcznie	Zautomatyzowany, czas skrócony o 5h
Teoretyczny	Możliwy, ale nieopłacalny	Teraz wykonywany automatycznie
Emergentny	Nigdy wcześniej nie brany pod uwagę	Nowe sposoby analizy, nowe decyzje, nowe pomysły

Tabela 1: Przechodzenie procesów przez kolejne poziomy rozwoju

Wniosek: Każda optymalizacja nie tylko redukuje czas trwania czynności, ale może również prowadzić do pojawienia się **nowych procesów**, które wcześniej były niemożliwe do wykonania lub nawet nie były brane pod uwagę.

5.4 Firma jako zbiór czynności i redukcja ich kosztów

Motywacja Firma może być postrzegana jako zbiór wykonywanych czynności, które składają się na procesy. Każda czynność generuje koszty wynikające z jej czasu trwania i częstotliwości wykonywania. Optymalizacja zbioru czynności sprowadza się zatem do analizy czynności i redukcji czasu ich trwania.

Struktura firmy jako zbioru czynności Formalnie można opisać firmę jako zbiór procesów:

$$F = \{P_1, P_2, \ldots, P_m\} \tag{101}$$

gdzie P_i to procesy składające się z czynności:

$$P_i = \{C_1, C_2, \ldots, C_n\} \tag{102}$$

Każda czynność C_i charakteryzuje się:

- $T(C_i)$ - czasem trwania,
- $K(C_i)$ - kosztem wykonania na jednostkę czasu,
- $f(C_i)$ - częstotliwością wykonywania.

Definicja różnicy czynności Optymalizacja procesu polega na zastąpieniu czynności C_i jej zoptymalizowaną wersją C_i', co można zapisać jako:

$$\Delta C_i = C_i - C_i' = (T(C_i) - T(C_i')) \times K(C_i) \times f(C_i) \tag{103}$$

gdzie ΔC_i oznacza oszczędność wynikającą ze skrócenia czasu trwania czynności.
Całkowity zysk z optymalizacji procesu P wynosi:

$$Zysk_P = \sum_{i=1}^{n} \Delta C_i \tag{104}$$

Przykład optymalizacji Rozważmy czynność C_x, która w starej wersji trwa 5 minut, jest wykonywana dwa razy w tygodniu i koszt roboczogodziny wynosi 7 €/h. Po optymalizacji czas czynności został zredukowany do 2 minut.

$$T(C_x) = 5 \text{ min} = \frac{5}{60} \text{ h} \tag{105}$$

$$T(C_x') = 2 \text{ min} = \frac{2}{60} \text{ h} \tag{106}$$

$$f(C_x) = 2 \times 52 = 104 \text{ razy w roku} \tag{107}$$

$$K(C_x) = 7 \text{ €/h} \tag{108}$$

Różnica kosztów wynosi:

$$\Delta C_x = \left(\frac{5}{60} - \frac{2}{60}\right) \times 7 \times 104 \tag{109}$$

$$\Delta C_x = \frac{3}{60} \times 7 \times 104 = 36.40 \text{ € rocznie.} \tag{110}$$

Jeżeli w firmie istnieje 50 takich czynności, to łączna oszczędność wynosi:

$$Zysk_P = 50 \times 36.40 = 1820 \text{ € rocznie.} \tag{111}$$

Podsumowanie Firma jako zbiór czynności może być optymalizowana poprzez redukcję czasu trwania operacji. Algebraiczna definicja różnicy czynności pozwala systematycznie analizować oszczędności wynikające z optymalizacji. Dzięki temu można zidentyfikować obszary o najwyższym potencjale redukcji kosztów i zwiększyć efektywność procesów.

5.5 Klasyfikacja oszczędności w optymalizacji czynności

Każda optymalizacja prowadzi do redukcji kosztów operacyjnych, jednak nie wszystkie oszczędności mają ten sam charakter. wyróżnić można trzy poziomy oszczędności w kontekście redukcji czasu czynności, zależne od stopnia wpływu na organizację.

5.5.1 Oszczędności realne – ciągłe zyski z optymalizacji istniejącego procesu

Definicja: Oszczędności wynikające z optymalizacji czynności, które były i nadal będą wykonywane.

Kluczowa cecha: Zysk z optymalizacji jest powtarzalny i cykliczny – występuje dopóki proces jest wykonywany okresowo.

Formalizacja:

$$\Delta C_{\text{realne}} = (T(C) - T(C')) \times K(C) \times f(C) \tag{112}$$

Gdzie:

- $T(C)$ – czas trwania czynności przed optymalizacją,
- $T(C')$ – czas trwania czynności po optymalizacji,
- $K(C)$ – koszt pracy na jednostkę czasu,
- $f(C)$ – częstotliwość wykonywania czynności.

Przykład: Optymalizacja zatwierdzania zamówień w systemie – zamiast ręcznego wpisywania danych, automatyczne wypełnianie pól w formularzu. Dzięki temu czas jednej operacji zmniejsza się z 5 minut do 2 minut. Efekt? Stała oszczędność na każdej transakcji, co skaluje się w całym systemie.

5.5.2 Oszczędności teoretyczne – umożliwienie wcześniej niepraktycznych działań

Definicja: Oszczędności wynikające z tego, że nowa metoda pozwala wykonywać czynności, które starą metodą byłyby zbyt czasochłonne i nieopłacalne.

Kluczowa cecha: Nie było wcześniejszej wersji tej czynności w systemie, ponieważ jej wykonanie było zbyt kosztowne lub wymagałoby nadmiernej ilości czasu.

Formalizacja:

$$\Delta C_{\text{teoretyczne}} = (T_{\text{teor}}(C) - T(C')) \times K(C) \times f_{\text{przewidywane}}(C) \tag{113}$$

Gdzie:

- $T_{\text{teor}}(C)$ – czas trwania czynności przy założeniu jej wykonania starą metodą (teoretyczny czas wykonania, gdyby była wykonywana ręcznie),
- $T(C')$ – czas trwania czynności po wdrożeniu nowej metody,
- $K(C)$ – koszt pracy na jednostkę czasu,
- $f_{\text{przewidywane}}(C)$ – oszacowana częstotliwość wykonywania nowej czynności po wdrożeniu.

Przykład: Automatyczna analiza statusów maszyn w systemie Syncos. Wcześniej analizowanie wydajności maszyn wymagało ręcznego raportowania, co było zbyt czasochłonne i nieopłacalne. Załóżmy, że:

- ręczne raportowanie zajmowałoby $T_{\text{teor}}(C) = 30$ minut,
- po wdrożeniu algorytmu analiza zajmuje $T(C') = 2$ minuty,
- koszt pracy wynosi $K(C) = 20$ EUR/h,
- przewidywana liczba analiz w miesiącu to $f_{\text{przewidywane}}(C) = 50$.

Oszczędność roczna wynosi:

$$\Delta C_{\text{teoretyczne}} = \left(\frac{30}{60} - \frac{2}{60}\right) \times 20 \times 50 \times 12 \tag{114}$$

$$\Delta C_{\text{teoretyczne}} = (0.5 - 0.0333) \times 20 \times 600 = 5600 \text{ EUR rocznie.} \tag{115}$$

Dzięki wdrożeniu możliwe jest wykonywanie analiz, które wcześniej byłyby zbyt czasochłonne, co przekłada się na oszczędność czasu i kosztów operacyjnych.

5.5.3 Emergentne oszczędności – nowe możliwości dzięki technologii

Definicja: Oszczędności wynikające z faktu, że nowe rozwiązanie stworzyło możliwość wykonywania czynności, które wcześniej były niemożliwe do przeprowadzenia.

Kluczowa cecha: Nie było nawet potrzeby optymalizacji wcześniej, bo dany proces nie istniał – nowe narzędzie umożliwia wykonanie działań, które wcześniej nie były w ogóle możliwe.

Formalizacja:

$$\Delta C_{\text{emergentne}} = \sum_{j} (T(C'_j) \times K(C_j) \times f(C_j)) + G_{\text{nowe}} \tag{116}$$

Gdzie:

- $T(C'_j)$ – czas nowej czynności, która stała się możliwa,
- $K(C_j)$ – koszt pracy na jednostkę czasu,
- $f(C_j)$ – częstotliwość wykonywania nowej czynności,
- G_{nowe} – wartość dodana wynikająca z czynności, które wcześniej były niemożliwe do wykonania.

Przykład: Dzięki systemowi BPOStream, firma może śledzić w czasie rzeczywistym przepływ części między obszarami, co nigdy wcześniej nie było możliwe.

- **Wcześniej** – brak narzędzi, brak możliwości monitorowania, zarządzanie zapasami było prowadzone na podstawie historycznych danych i fizycznych kontroli.
- **Teraz** – automatyczne monitorowanie buforów, optymalizacja zatwierdzeń i dynamiczna alokacja zasobów na podstawie bieżących stanów magazynowych.

Efekt? Powstają nowe procesy, które generują wartość dodaną i redukują ukryte straty operacyjne.

Podział oszczędności w tym przypadku:

- **Realne oszczędności:** Skrócenie czasu spędzonego przez pracowników na ręcznej inwentaryzacji buforów.
- **Teoretyczne oszczędności:** Możliwość bieżącej analizy danych magazynowych, co wcześniej było zbyt czasochłonne.
- **Emergentne oszczędności:** Nowe metody dynamicznej alokacji zasobów na podstawie bieżących danych, co wcześniej nie było możliwe.

5.5.4 Podsumowanie: Hierarchia oszczędności czasu

Rodzaj oszczędności	Co optymalizuje?	Jakie efekty?
Realne	Procesy, które już istnieją	Skrócenie czasu istniejących czynności = stała oszczędność
Teoretyczne	Procesy, które były zbyt czasochłonne	Czynności, które kiedyś były nieopłacalne, stają się możliwe
Emergentne	Procesy, które wcześniej nie istniały	Nowe rozwiązania umożliwiają czynności, których nie dało się wykonać wcześniej + wartość emergentna (G_{nowe})

Tabela 2: Hierarchia oszczędności w optymalizacji procesów

Klasyfikacja oszczędności

1. Oszczędności realne (S_r)

Zmniejszenie kosztu czasu pracy poprzez eliminację lub redukcję czasu trwania czynności.

$$S_r = T_{\text{przed}} - T_{\text{po}} \tag{117}$$

gdzie:

- T_{przed} – czas trwania czynności przed wdrożeniem,
- T_{po} – czas po wdrożeniu.

2. Oszczędności potencjalne (skalowalne) (S_p)

Eliminacja wzrostu kosztów przy zwiększającej się liczbie powtarzalnych procesów.

$$S_p = n \cdot (T_{\text{przed}} - T_{\text{po}}) \tag{118}$$

gdzie:

- n – liczba instancji procesu po rozbudowie skali.

3. Oszczędności pośrednie (S_i)

Redukcja kosztów wynikająca z lepszego dostępu do danych i szybszego podejmowania decyzji, co wpływa na inne czynniki operacyjne.

$$S_i = f(D, Q, R) \tag{119}$$

gdzie:

- D – skrócony czas dostępu do danych,
- Q – poprawa jakości decyzji,
- R – wpływ na redukcję strat, np. zmniejszenie odpadu.

Przykłady oszczędności potencjalnych

Przykład 1: Automatyzacja porządkowania danych e-learningowych

Opis: Wcześniej porządkowanie danych z jednego kursu e-learningowego trwało 2 godziny i było wykonywane 2 razy w miesiącu. Po wdrożeniu automatycznego systemu dane są przetwarzane natychmiast po wrzuceniu pliku Excel.

Oszczędność realna

Dane:

- $T_{\text{przed}} = 2$ godziny na jedno wykonanie,
- $T_{\text{po}} \approx 0$ godzin na jedno wykonanie,
- $f = 2$ (liczba wykonanych operacji miesięcznie).

Oszczędność realna (czasowa):

$$S_r = f \cdot (T_{\text{przed}} - T_{\text{po}}) = 2 \cdot (2 - 0) = 4 \text{ godziny miesięcznie} \tag{120}$$

Oszczędność realna (pieniężna): Przyjmując stawkę roboczogodziny na poziomie 12 euro:

$$S_r^{\text{EUR}} = 4 \cdot 12 \cdot 12 = 576 \text{ euro rocznie} \tag{121}$$

Oszczędność potencjalna

Dane:

- $n = 10$ (planowana liczba kursów e-learningowych w przyszłości).

Oszczędność potencjalna (czasowa):

$$S_p = n \cdot S_r = 10 \cdot 4 = 40 \text{ godzin miesięcznie} \tag{122}$$

Oszczędność potencjalna (pieniężna):

$$S_p^{\text{EUR}} = 10 \cdot 576 = 5760 \text{ euro rocznie} \tag{123}$$

W przypadku realizacji 10 kursów e-learningowych starą metodą, czas wymagany na opracowanie wszystkich kursów wynosiłby:

$$S_{\text{old}} = n \cdot S_r = 10 \cdot 4 \cdot 12 = 480 \text{ godzin rocznie} \tag{124}$$

Przy stawce **12 euro za godzinę** roczny koszt opracowania tych kursów wynosiłby:

$$C_{\text{old}} = 480 \cdot 12 = 5760 \text{ euro rocznie} \tag{125}$$

W skali organizacyjnej oznacza to, że praca ta wymagałaby **około 0,24 etatu** pełnoetatowego pracownika (przy 8 godzinach dziennie i 250 dniach roboczych w roku):

$$N_{\text{pracowników}} = \frac{C_{\text{old}}}{12 \times 8 \times 250} = 0.24 \tag{126}$$

Dzięki wdrożonemu rozwiązaniu **firma eliminuje konieczność alokowania dodatkowego pracownika** do tych zadań, co umożliwia efektywniejsze wykorzystanie zasobów ludzkich i automatyzację procesu bez dodatkowych kosztów operacyjnych.

Przykład 2: Pareto odpadu w czasie rzeczywistym

Opis: Wcześniej analiza Pareto odpadu wymagała zbierania danych i trwała 30 minut dla jednej części. Teraz analiza jest generowana natychmiast po wybraniu części z listy, co oznacza, że w czasie jednej wcześniejszej analizy można teraz wykonać wielokrotność tej liczby.

Oszczędność realna

Dane:

- $T_{\text{przed}} = 30$ minut (0.5 godziny) na jedną analizę,
- $T_{\text{po}} \approx 0$ godzin,
- $n_{2024} = 50$ (liczba analiz wykonanych w roku 2024).

Oszczędność realna (czasowa):

$$S_r = n_{2024} \cdot (T_{\text{przed}} - T_{\text{po}}) = 50 \cdot (0.5 - 0) = 25 \text{ godzin rocznie} \tag{127}$$

Oszczędność realna (pieniężna): Przyjmując stawkę roboczogodziny na poziomie 12 euro:

$$S_r^{\text{EUR}} = 25 \cdot 12 = 300 \text{ euro rocznie} \tag{128}$$

Oszczędność potencjalna

Opis: Wcześniej w czasie jednej analizy (0.5 godziny) można było wykonać tylko jedną analizę. Teraz w tym samym czasie można wykonać nawet 20 analiz.

Dane:

- $k = 20$ (możliwa liczba analiz w czasie jednej wcześniejszej analizy).

Oszczędność potencjalna (czasowa):

$$S_p = k \cdot S_r = 20 \cdot 25 = 500 \text{ godzin rocznie} \tag{129}$$

Oszczędność potencjalna (pieniężna):

$$S_p^{\text{EUR}} = 500 \cdot 12 = 6000 \text{ euro rocznie} \tag{130}$$

Oszczędność związana z potencjałem decyzyjnym

Opis: Wcześniej jedna analiza Pareto trwała 30 minut, co ograniczało liczbę analiz, jakie można było wykonać w danym czasie. Teraz pracownik może wykonać do 100 analiz dziennie.

Dane:

- $A_{\text{new}} = 100$ (liczba analiz możliwych do wykonania dziennie nową metodą),
- $D = 250$ (liczba dni roboczych w roku),
- $A_{\text{year}} = A_{\text{new}} \cdot D = 100 \times 250 = 25000$ (liczba analiz rocznie nową metodą).

Oszczędność czasowa starej metody:

$$S_d = A_{\text{year}} \cdot T_{\text{przed}} = 25000 \cdot 0.5 = 12500 \text{ godzin rocznie} \tag{131}$$

Oszczędność pieniężna:

$$S_d^{\text{EUR}} = 12500 \cdot 12 = 150000 \text{ euro rocznie} \tag{132}$$

Odpowiada to wynagrodzeniu 6,25 pełnoetatowych pracowników (przy stawce 12 euro/godz., 8 godzin dziennie, 250 dni roboczych rocznie):

$$N_{\text{pracowników}} = \frac{S_d^{\text{EUR}}}{12 \times 8 \times 250} = 6.25 \tag{133}$$

Dzięki wdrożonemu rozwiązaniu zwiększyła się zdolność decyzyjna firmy. Można teraz wykonywać czynność do której wcześniej potrzebne byłoby 6 pracowników.

6 Statyczny model priorytetyzacji wysyłek

6.1 Opis modelu

Statyczny model priorytetyzacji pozwala na obliczenie wartości priorytetu dla danego detalu w zależności od trzech kluczowych czynników:

- **Terminu wysyłki** – określa liczbę dni do zaplanowanej wysyłki.
- **Ilości brakujących części** – różnica między zapotrzebowaniem a stanem magazynowym.
- **Historii stanów magazynowych** – umożliwia oszacowanie tempa dopływu części do magazynu końcowego.

6.2 Definicje zmiennych

Niech:

- $X_{\text{wysyłka}}$ – liczba części wymaganych do realizacji danej wysyłki,
- S_{magazyn} – aktualny stan części w magazynie wysyłkowym,
- X_{nadmiar} – liczba części, które pozostają po zabezpieczeniu bieżącej wysyłki,
- $X_{\text{brakujące}}$ – liczba brakujących części do realizacji wysyłki,
- $T_{\text{wysyłka}}$ – liczba dni do planowanej wysyłki,
- $F_{\text{częstotliwość}}$ – średnia liczba części pojawiających się dziennie w magazynie końcowym.

6.3 Obliczenie ilości brakujących części

$$X_{\text{brakujące}} = \max(0, X_{\text{wysyłka}} - S_{\text{magazyn}}) \tag{134}$$

Jeśli stan magazynu S_{magazyn} jest większy niż wymagana ilość $X_{\text{wysyłka}}$, to mamy nadmiar części:

$$X_{\text{nadmiar}} = S_{\text{magazyn}} - X_{\text{wysyłka}} \tag{135}$$

6.4 Przenoszenie nadmiaru części na kolejną wysyłkę

Jeśli dla bieżącej wysyłki mamy nadmiar części $X_{\text{nadmiar}} > 0$, możemy przenieść go na następną wysyłkę tej samej części:

$$S_{\text{kolejna wysyłka}} = S_{\text{kolejna wysyłka}} + X_{\text{nadmiar}} \tag{136}$$

Dzięki temu kolejna wysyłka może być częściowo zabezpieczona już teraz.

6.5 Uwzględnienie historycznych danych o dopływie części

Możemy oszacować średnie tempo pojawiania się części w magazynie na podstawie historycznych danych:

$$F_{\text{częstotliwość}} = \frac{\sum X_{\text{historia}}}{T_{\text{historia}}} \tag{137}$$

Gdzie:

- X_{historia} – liczba części, które pojawiły się w magazynie w analizowanym okresie,
- T_{historia} – liczba dni w analizowanym okresie.

Na tej podstawie możemy oszacować, ile części pojawi się w magazynie przed wysyłką:

$$X_{\text{prognoza}} = F_{\text{częstotliwość}} \cdot T_{\text{wysyłka}} \tag{138}$$

6.6 Obliczenie priorytetu wysyłki

Priorytet wysyłki obliczamy jako:

$$P_{\text{wysyłka}} = \frac{X_{\text{brakujące}}}{T_{\text{wysyłka}} + 1} \times \frac{1}{F_{\text{częstotliwość}} + 1} \tag{139}$$

Gdzie:

- Im większa liczba brakujących części $X_{\text{brakujące}}$, tym większy priorytet.
- Im mniej dni do wysyłki $T_{\text{wysyłka}}$, tym większy priorytet.
- Im częściej części pojawiają się w historii $F_{\text{częstotliwość}}$, tym niższy priorytet.

6.7 Interpretacja modelu

Model ten ma charakter **statyczny**, co oznacza, że oblicza priorytet na podstawie aktualnego stanu w chwili t. Wartości obliczane są na podstawie aktualnych danych, nie uwzględniając przyszłych dynamicznych zmian. Dzięki temu model może być używany do szybkiej oceny sytuacji na dany moment oraz do priorytetyzacji działań operacyjnych.

6.8 Obliczenie pokrycia zapotrzebowania

Pokrycie zapotrzebowania określa, w jakim stopniu obecne tempo dopływu części jest w stanie pokryć zapotrzebowanie wynikające z nadchodzącej wysyłki. Definiujemy je jako:

$$C_{\text{pokrycie}} = \frac{F_{\text{częstotliwość}} \times T_{\text{wysyłka}}}{X_{\text{brakujące}}} \times 100\% \tag{140}$$

Gdzie:

- C_{pokrycie} – procentowe pokrycie zapotrzebowania przed wysyłką,
- $F_{\text{częstotliwość}}$ – średnie tempo pojawiania się części w magazynie końcowym,
- $T_{\text{wysyłka}}$ – liczba dni do wysyłki,
- $X_{\text{brakujące}}$ – liczba brakujących części do realizacji wysyłki.

6.8.1 Interpretacja pokrycia

- $C_{\text{pokrycie}} = 100\%$ oznacza, że dopływ części do magazynu wystarczy dokładnie na pokrycie zapotrzebowania w dniu wysyłki.
- $C_{\text{pokrycie}} > 100\%$ oznacza, że tempo dopływu części jest wyższe niż wymagane – w takim przypadku możemy spodziewać się nadmiaru części w magazynie.
- $C_{\text{pokrycie}} < 100\%$ oznacza, że nie jesteśmy w stanie pokryć zapotrzebowania na czas, co oznacza konieczność zwiększenia priorytetu dla danej części.

6.8.2 Zależność między pokryciem a priorytetem

Pokrycie może być wykorzystane jako wskaźnik pomocniczy w priorytetyzacji. Priorytet jest odwrotnie proporcjonalny do pokrycia:

$$P_{\text{wysyłka}} \sim \frac{1}{C_{\text{pokrycie}}} \tag{141}$$

Im niższe pokrycie, tym wyższy priorytet, co oznacza, że części, których zapotrzebowanie nie jest wystarczająco pokryte, będą traktowane jako bardziej pilne.

—

7 Rozważania

7.0.1 Analiza Przypadków w Rodzinach Parametrycznych

Rodziny parametryczne obejmują procesy, które mają tę samą strukturę relacyjną $\mathcal{R}$ oraz funkcje przejścia $\mathcal{T}$, różniąc się jedynie konfiguracją przestrzeni stanów.

Przypadek 1: Zmiana Wartości w Przestrzeni Stanów Jeśli zmieniają się tylko wartości numeryczne lub kategoryczne w przestrzeniach stanów S_i, odwzorowanie g_i jest prostą transformacją, która nie zmienia struktury ani reprezentacji przestrzeni stanów. Zapewnia to, że proces pozostaje izomorficzny względem swojej pierwotnej formy:

$$g_i(s) = s + \Delta s, \quad \forall s \in S_i. \tag{142}$$

Przykład: Dostosowanie parametru numerycznego, takiego jak czas cyklu w przestrzeni stanów maszyny:

$$g_{\text{maszyna}}(s) = s + \Delta(\text{czas\_cyklu}), \tag{143}$$

gdzie $\Delta(\text{czas\_cyklu})$ reprezentuje stałą lub czynnik skalujący.

Zachowane Właściwości:

- Relacje $\mathcal{R}$ pozostają niezmienione.
- Topologia procesu jest zachowana.
- Funkcje przejścia $\mathcal{T}$ pozostają spójne względem zmienionych wartości.

Takie zmiany są szczególnie przydatne w analizie wrażliwości, gdzie zachowanie dynamiczne procesu jest testowane w szerokim zakresie wartości parametrów.

7.0.2 Wpływ Zmian w Reprezentacji Przestrzeni Stanów

Zmiany w reprezentacji przestrzeni stanów S_i mogą prowadzić do modyfikacji dziedziny funkcji przejścia $\mathcal{T}$. Jest to szczególnie istotne w przypadku dekompozycji algebraicznej, gdzie parametr zostaje przedefiniowany lub podzielony na wiele komponentów. Takie zmiany mają bezpośredni wpływ na zachowanie i strukturę systemu.

Przypadek 1: Bijektywna Dekompzycja Przestrzeni Stanów W tym przypadku przestrzeń stanów S_i zostaje przekształcona przy zachowaniu bijekcyjnego odwzorowania na oryginalną przestrzeń. Nowa przestrzeń stanów S_i' zachowuje jednoznaczną korespondencję z S_i, co zapewnia dobrze zdefiniowaną i spójną dziedzinę funkcji przejścia.

- **Definicja:** Parametr $p \in S_i$ zostaje zdekomponowany na $p_1, p_2, \ldots, p_k$ z deterministyczną relacją:

  $$p = f(p_1, p_2, \ldots, p_k), \tag{144}$$

  a nowa przestrzeń stanów przyjmuje postać:

  $$S_i' = \{(p_1, p_2, \ldots, p_k, s_2, \ldots, s_n) \mid f(p_1, p_2, \ldots, p_k) = p,\ s \in S_i\}. \tag{145}$$

  Aby zapewnić bijekcję, S_i' zostaje ograniczona do podprzestrzeni S_i'', gdzie:

  $$\forall s \in S_i, \exists! s' \in S_i'',\ \text{takie, że } g_i(s) = s'. \tag{146}$$

- **Zmiany w Funkcjach Przejścia:** Każda funkcja przejścia T_i zostaje dostosowana do działania w nowej dziedzinie S_i'. Na przykład funkcja w oryginalnej przestrzeni:

  $$T_i(s, i) = s + i, \tag{147}$$

  jest przekształcana na:

  $$T_i'((p_1, p_2), i) = (p_1, p_2 + i),\ \text{gdzie } p_1 + p_2 = p. \tag{148}$$

- **Implikacje:**
  - **Struktura Relacyjna:** Struktura relacyjna $\mathcal{R}$ pozostaje nienaruszona, co zapewnia zachowanie interakcji między obiektami w systemie.

- **Izomorfizm:** Oryginalny proces $\mathcal{P} = (\mathcal{O}, \mathcal{R}, \mathcal{T})$ oraz przekształcony proces $\mathcal{P}' = (\mathcal{O}, \mathcal{R}, \mathcal{T}')$ są izomorficzne:

$$\mathcal{P} \cong \mathcal{P}'. \tag{149}$$

  Wynika to z faktu, że zmiany mają charakter czysto reprezentacyjny i nie wpływają na logikę ani topologię procesu.

- **Rodzina Parametryczna:** System pozostaje w ramach rodziny parametrycznej $\mathcal{F}_{\text{parametryczna}}$, ponieważ transformacja zachowuje równoważność konfiguracji stanów i logikę przejść. Bijekcja między S_i a S_i' gwarantuje, że zmodyfikowany proces można traktować jako zmiana reprezentacji oryginalnego.

- **Analiza i Optymalizacja:**
  - Bijekcyjna dekompozycja umożliwia eksplorację alternatywnych reprezentacji parametrów w ramach tej samej struktury.
  - Może to być szczególnie przydatne w analizie wrażliwości lub optymalizacji, gdzie reprezentacja zmiennych stanu wpływa na łatwość obliczeń lub interpretowalność, ale nie zmienia inherentnych właściwości systemu.

Przypadek 2: Niebijektywna dekompozycja przestrzeni stanów W tym scenariuszu dekompozycja S_i wprowadza redundancję z powodu braku dodatkowych ograniczeń, co skutkuje niebijektywnym odwzorowaniem pomiędzy S_i a S_i'. Chociaż struktura relacyjna $\mathcal{R}$ oraz zbiór funkcji przejścia $\mathcal{T}$ pozostają ważne, dynamika systemu może wykazywać inne zachowania z powodu rozszerzonej domeny.

- **Definicja:** Parametr $p \in S_i$ jest dekomponowany bez dodatkowych ograniczeń, co skutkuje:

$$S_i' = \{(p_1, p_2) \mid p_1 + p_2 = p,\ s \in S_i\}. \tag{150}$$

  Nowa domena S_i' jest większa niż oryginalna przestrzeń S_i, co oznacza rozszerzenie przestrzeni parametrów. Przykładowo:

$$S_i' = \{(p_1, p_2) \mid p_1 + p_2 = 6\}, \tag{151}$$

  gdzie S_i' zawiera kombinacje takie jak $(3,3), (2,4), (4,2), \ldots$.

- **Zmiany w funkcjach przejścia:** Funkcje przejścia T_i są dostosowane do działania w rozszerzonej domenie S_i', co prowadzi do potencjalnie zmienionej dynamiki. Przykładowo:

$$T_i(s, i) = s + i \tag{152}$$

  staje się:

$$T_i'((p_1, p_2), i) = (p_1, p_2 + i), \text{ gdzie } p_1 + p_2 = p. \tag{153}$$

- **Wpływ na dynamikę:** Chociaż proces pozostaje operacyjny, rozszerzona domena przestrzeni stanów pozwala na nowe trajektorie i konfiguracje w ewolucji systemu. Zmiany te nie zakłócają procesu, ale mogą wprowadzić większą elastyczność lub zmienność w jego zachowaniu.

Dodatkowe uwagi na temat niebijektywnej dekompozycji:

- **Redundancja i nadmiarowa reprezentacja:** Redundancja wprowadzona przez S_i' oznacza, że jeden stan w S_i jest teraz reprezentowany przez wiele stanów w S_i'. Ta redundancja niekoniecznie dodaje nową informację, ale zwiększa stopnie swobody w reprezentacji, co może komplikować analizę zachowania systemu.

- **Przynależność do rodziny funkcjonalnej:** - Proces należy do rodziny funkcjonalnej $\mathcal{F}_{\text{functional}}$, ponieważ dekompozycja wprowadza zmodyfikowaną domenę dla funkcji przejścia $\mathcal{T}'$, umożliwiając eksplorację nowych wzorców dynamicznych. - Brak bijekcji uniemożliwia klasyfikację w rodzinie parametrycznej $\mathcal{F}_{\text{parametric}}$, ponieważ redundancja w S_i' narusza wymóg jednoznacznej korespondencji.

- **Wpływ na logikę przejść:** - Funkcje przejścia T_i' muszą być zdefiniowane na nowo, aby działały w rozszerzonej przestrzeni stanów, co może prowadzić do nowych zachowań niedostępnych w oryginalnej przestrzeni S_i. - Przeprojektowanie funkcji może stworzyć możliwości elastyczności systemu, ale wymaga dokładnej walidacji w celu zapewnienia spójności.
- **Dynamiczne implikacje redundancji:** - Rozszerzona przestrzeń stanów pozwala na wiele trajektorii lub reprezentacji tego samego procesu, co może wprowadzać zmienność lub nawet dwuznaczność w obserwowanym zachowaniu systemu. - Przykładowo, w przypadku gdy $S_i' = \{(p_1, p_2) \mid p_1 + p_2 = 6\}$, system może preferować różne dekompozycje w zależności od czynników zewnętrznych lub kryteriów optymalizacji.
- **Porównanie z bijektywną dekompozycją:** - W przeciwieństwie do bijektywnej dekompozycji (Przypadek 1), gdzie transformacja ma charakter wyłącznie reprezentacyjny i nie wpływa na logikę systemu, niebijektywna dekompozycja ma potencjał wprowadzenia nowych dynamik poprzez rozszerzenie domeny możliwych konfiguracji. - Ta różnica sprawia, że Przypadek 2 jest z natury bardziej złożony i lokuje go w zakresie zmian funkcjonalnych.

Przypadek 3: Dodanie niezależnego parametru do przestrzeni stanów W tym scenariuszu przestrzeń stanów S_i zostaje rozszerzona poprzez wprowadzenie nowego, niezależnego parametru q. Ta modyfikacja skutkuje rozszerzoną przestrzenią stanów S_i' i może wymagać aktualizacji funkcji przejścia T_i. Dodany parametr nie zmienia istniejących relacji $\mathcal{R}$ ani liczności zbioru obiektów $\mathcal{O}$, ale wprowadza nowe wymiary do dynamiki procesu.

- **Definicja:** Nowy parametr $q \in Q$ zostaje dodany do przestrzeni stanów S_i, co prowadzi do:

  $$S_i' = S_i \times Q. \tag{154}$$

  Parametr q jest niezależny od istniejących parametrów w S_i, a nowa przestrzeń stanów przyjmuje postać:

  $$S_i' = \{(s, q) \mid s \in S_i, q \in Q\}. \tag{155}$$

- **Zmiany w funkcjach przejścia:** Jeśli q wpływa na dynamikę systemu, funkcje przejścia T_i muszą zostać dostosowane do uwzględnienia nowego parametru. Przykładowo:

  $$T_i(s, i) = s + i \tag{156}$$

  staje się:

  $$T_i'(s, q, i) = s + i + f(q), \tag{157}$$

  gdzie $f(q)$ modeluje wpływ q na dynamikę systemu.

- **Wpływ na dynamikę:** - Struktura relacyjna $\mathcal{R}$ pozostaje niezmieniona. - Nowy parametr q wprowadza dodatkową elastyczność w zachowaniu systemu, rozszerzając przestrzeń stanów i umożliwiając nowe konfiguracje. - Funkcje przejścia T_i mogą wykazywać rozszerzoną lub zmodyfikowaną dynamikę w zależności od roli q.
- **Przynależność do rodziny funkcjonalnej:** Proces $\mathcal{P}'$ pozostaje w rodzinie funkcjonalnej $\mathcal{F}_{\text{functional}}$, ponieważ strukturalne relacje $\mathcal{R}$ są zachowane, a zmiany ograniczają się do przestrzeni stanów i funkcji przejścia.

Wpływ dodania niezależnych parametrów:

- **Rozszerzona przestrzeń stanów:** Dodanie q zwiększa wymiarowość S_i, umożliwiając bardziej szczegółowe modelowanie dynamiki systemu.
- **Zwiększone możliwości analityczne:** Nowy parametr dostarcza możliwości analizy i optymalizacji systemu w odniesieniu do q.

- **Elastyczność dynamiczna:** Zmodyfikowane funkcje przejścia T_i' umożliwiają eksplorację nowych zachowań dynamicznych i konfiguracji.
- **Zachowanie integralności relacyjnej:** Struktura relacyjna $\mathcal{R}$ pozostaje nienaruszona, co zapewnia spójność w systemie.

—