

Piotr Chrzastowski-Wachtel

Sieci Petriego,
jako narzędzie do opisu
przepływów pracy
(workflows)

Przepływy pracy

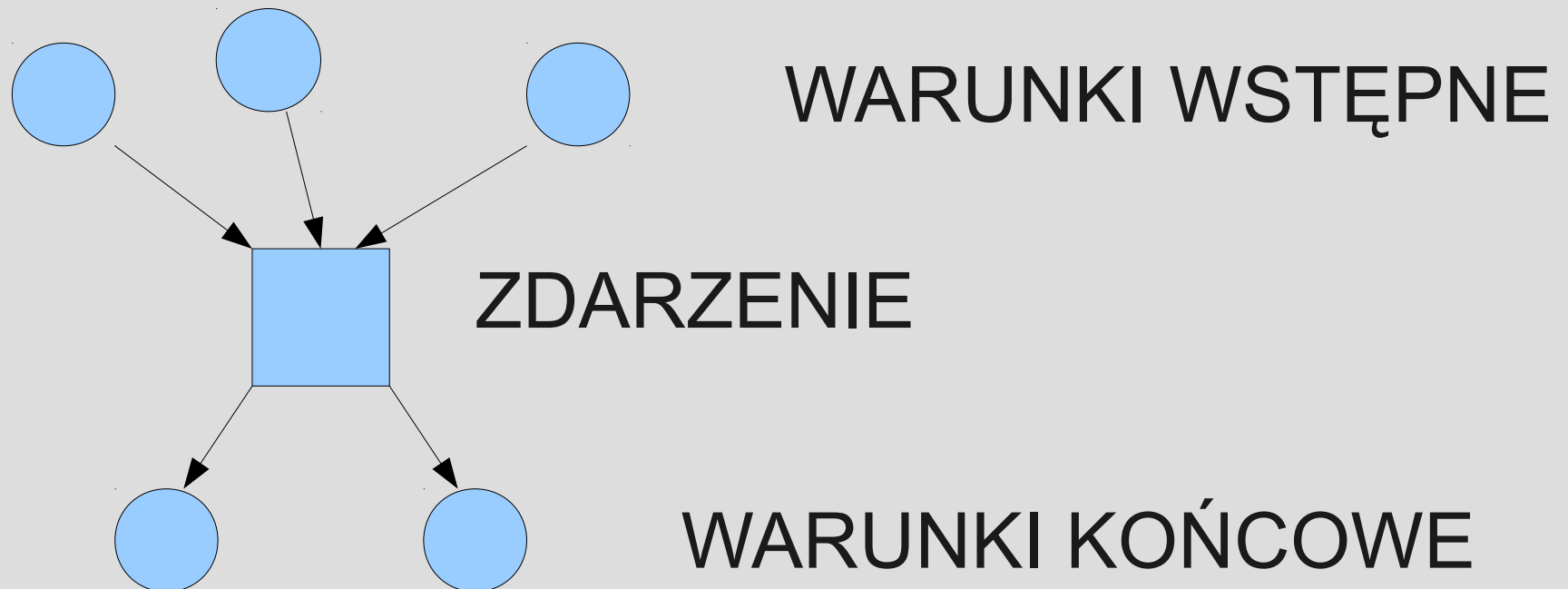
- Opisują przebieg wykonania przypadków biznesowych
- Zazwyczaj odzwierciedlają typowe, powtarzalne scenariusze
- Ukazują zależności czasowe między poszczególnymi elementami scenariusza
- Umożliwiają półautomatyczne zarządzanie projektem
- Pozwalają unaocznic bieżący stan projektu i prognozy co do jego zakończenia.

Sieci Petriego

- Model współbieżności zaproponowany przez C.A.Petriego w 1962 r.
- Najpopularniejszy model do przedstawiania rozmaitych procesów, kiedy wiele zdarzeń zachodzi w tym samym czasie, np.
 - systemy produkcyjne (flexible manufacturing)
 - przepływy pracy
 - protokoły komunikacyjne
- Ponad 10 tys. prac dotyczących metod analitycznych i zastosowań

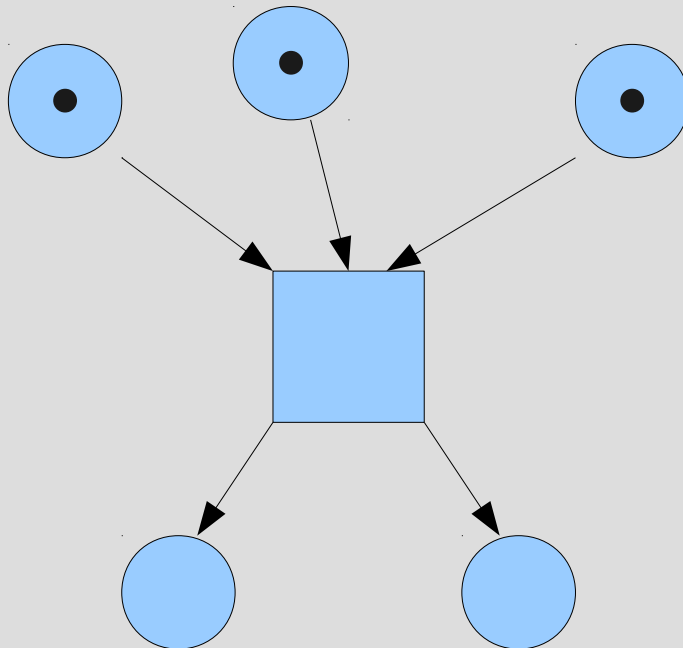
Podstawowy pomysł

- Pojedyncze zdarzenie ma pewne warunki wstępne, które muszą być spełnione i warunki wyjściowe, które staną się spełnione, gdy zdarzenie zostanie wykonane



Zajście zdarzenia - początek

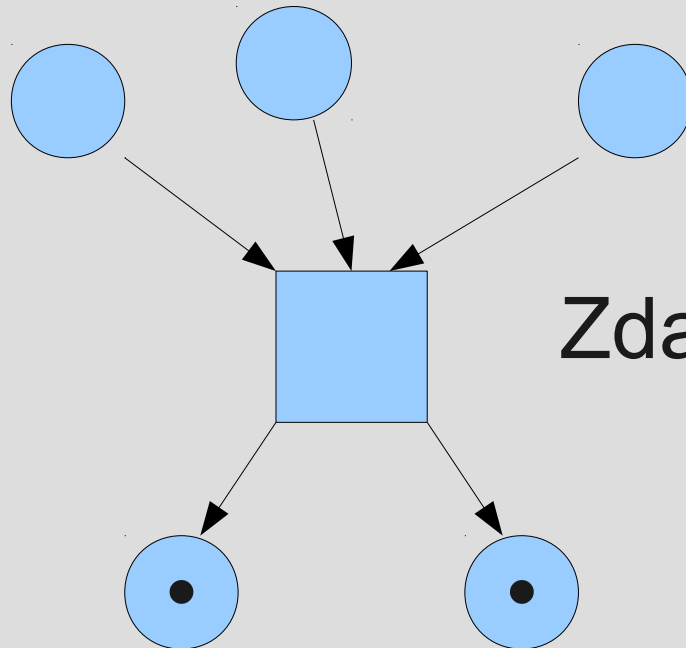
- Jak działa pojedyncze zdarzenie?
- Aby zadziałać musi być aktywowane, czyli na jego wejściach powinny być pionki, a wyjścia powinny być puste



Zdarzenie aktywne

Zajście zdarzenia - koniec

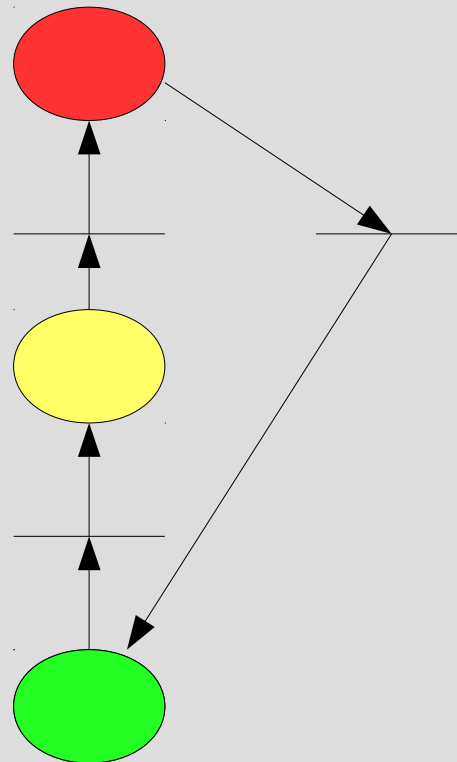
- Jak działa pojedyncze zdarzenie?
- Aby zadziałać musi być aktywowane, czyli na jego wejściach powinny być pionki, a wyjścia powinny być puste



Zdarzenie, które właśnie zaszło

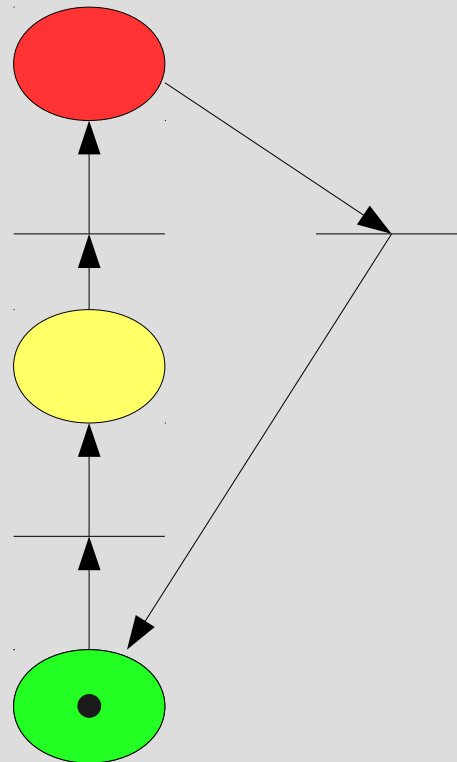
Sieci warunkowo-zdarzeniowe

- Sieci powstałe z połączenia za pomocą miejsc takich zdarzeń nazywamy sieciami warunkowo-zdarzeniowymi.



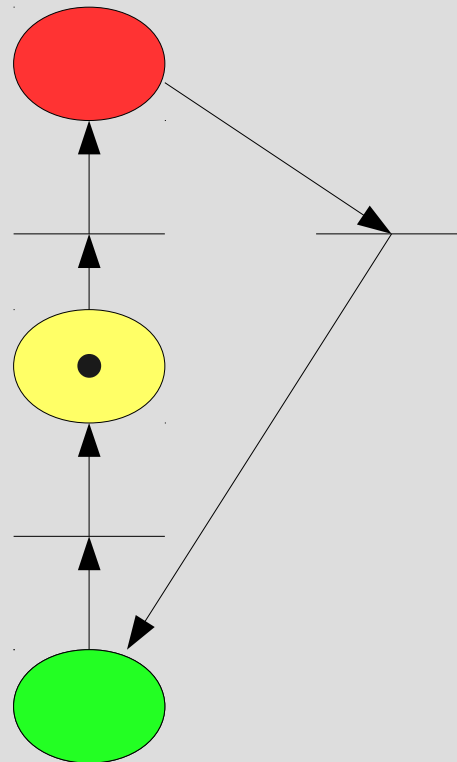
Sieci warunkowo-zdarzeniowe

- Sieci powstałe z połączenia za pomocą miejsc takich zdarzeń nazywamy sieciami warunkowo-zdarzeniowymi.



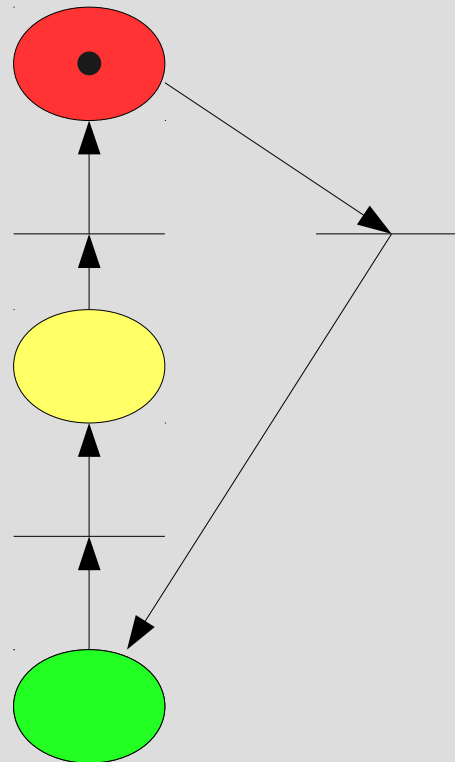
Sieci warunkowo-zdarzeniowe

- Sieci powstałe z połączenia za pomocą miejsc takich zdarzeń nazywamy sieciami warunkowo-zdarzeniowymi.



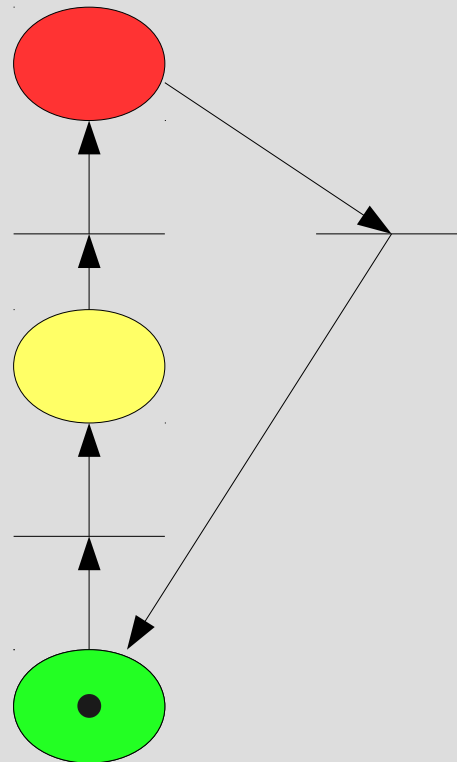
Sieci warunkowo-zdarzeniowe

- Sieci powstałe z połączenia za pomocą miejsc takich zdarzeń nazywamy sieciami warunkowo-zdarzeniowymi.



Sieci warunkowo-zdarzeniowe

- Sieci powstałe z połączenia za pomocą miejsc takich zdarzeń nazywamy sieciami warunkowo-zdarzeniowymi.

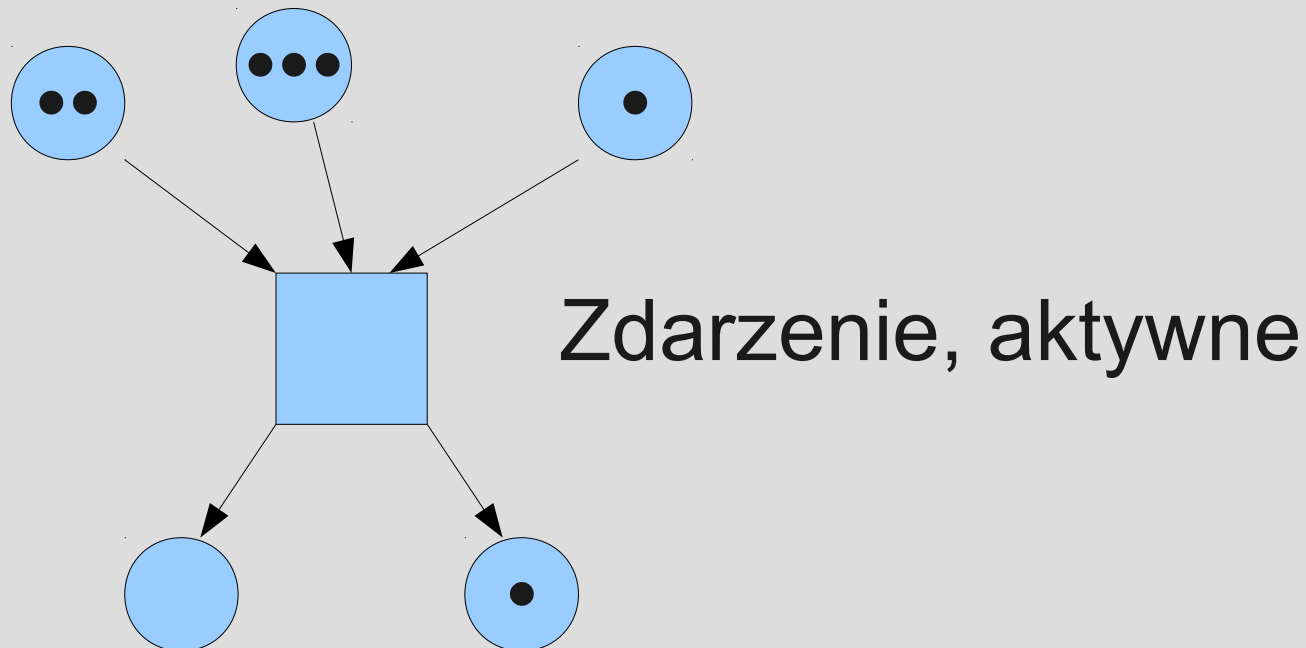


Sieci warunkowo-zdarzeniowe

- Przedstawione sieci obrazują zmienianie się warunków logicznych (lokalnych stanów)
- Pionek w miejscu odpowiadającym warunkowi oznacza jego spełnienie; brak pionka - jego fałszywość.
- Liczba konfiguracji osiągalnych z początkowego rozstawienia pionków jest skończona i stanowi przestrzeń wszystkich możliwych stanów, jakie system może przyjąć.

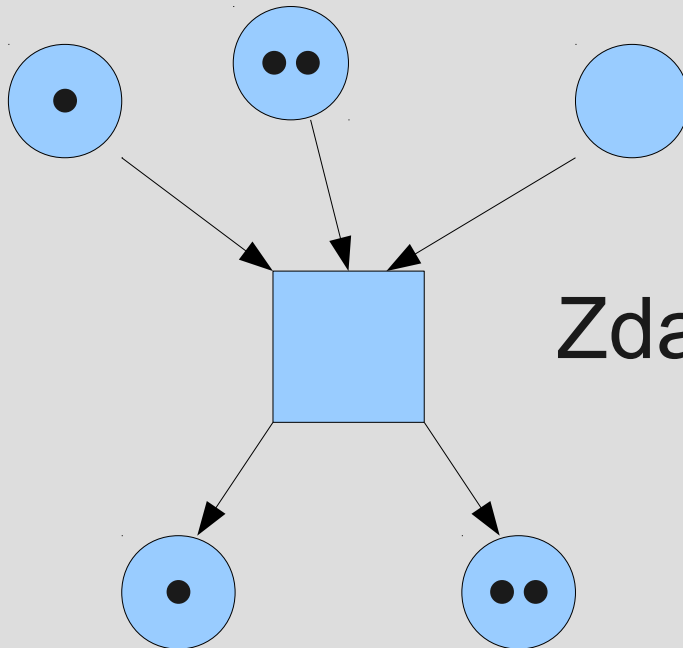
Alternatywna wersja - więcej pionków

- Jak działa pojedyncze zdarzenie?
- Aby zadziałać musi być aktywowane, czyli na jego wejściach powinny być pionki. Tym razem o wyjściach nic nie zakładamy



Alternatywna wersja - więcej pionków

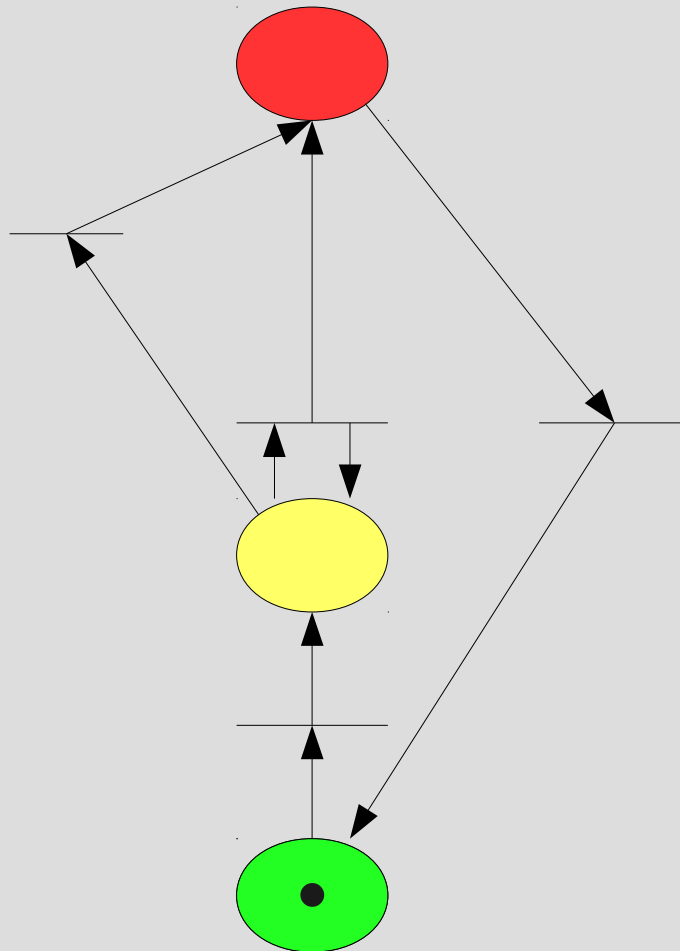
- Jak działa pojedyncze zdarzenie?
- Aby zadziałać musi być aktywowane, czyli na jego wejściach powinny być pionki. Tym razem o wyjściach nic nie zakładamy



Zdarzenie, które właśnie zaszło

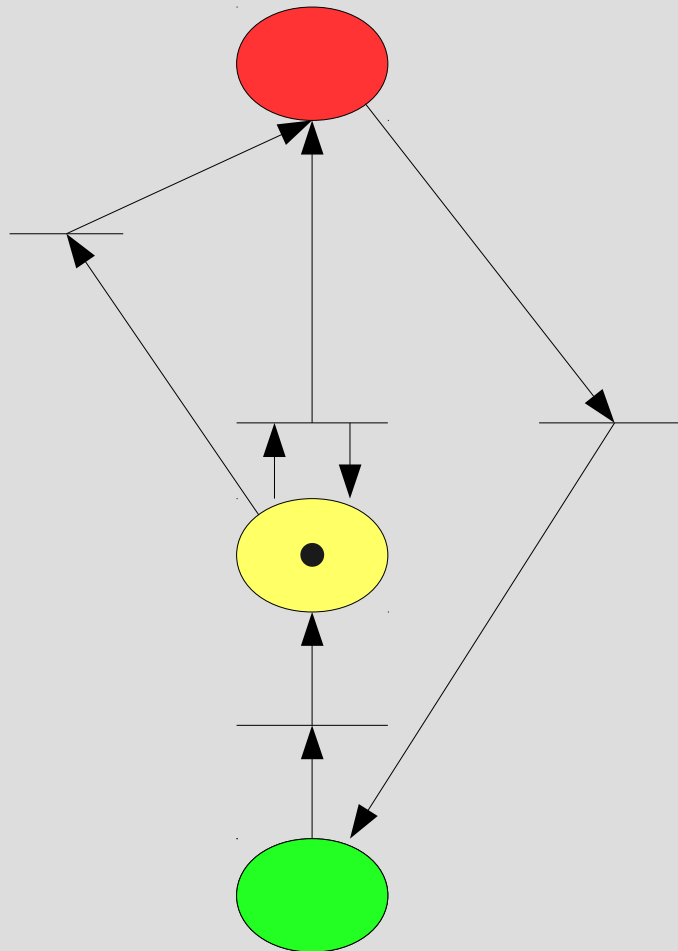
Modelowanie bardziej złożonych sytuacji

- Jak oddać właściwy porządek zmiany świateł?



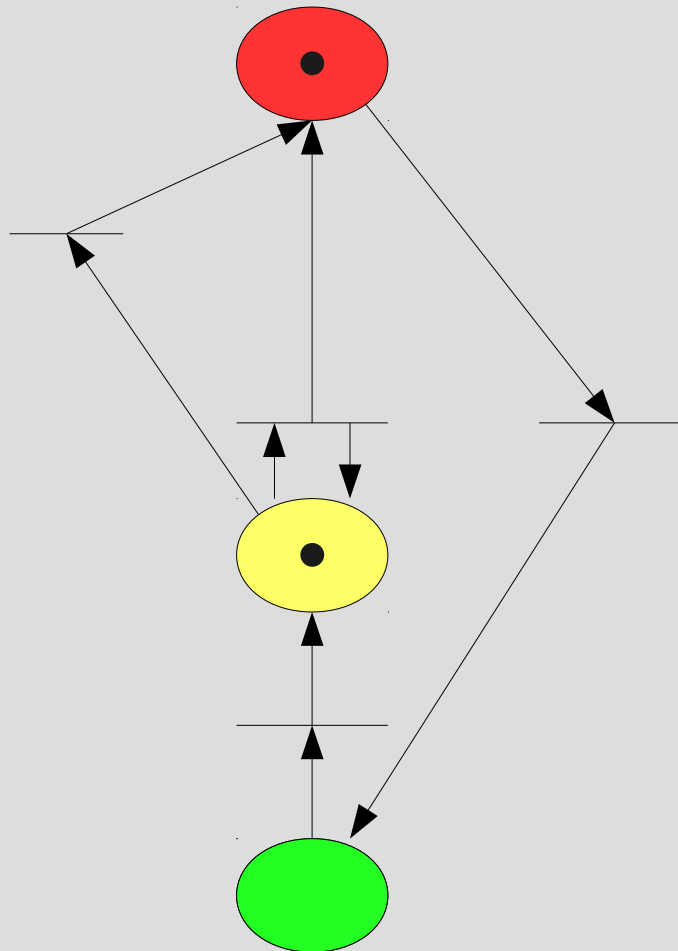
Modelowanie bardziej złożonych sytuacji

- Jak oddać właściwy porządek zmiany świateł?



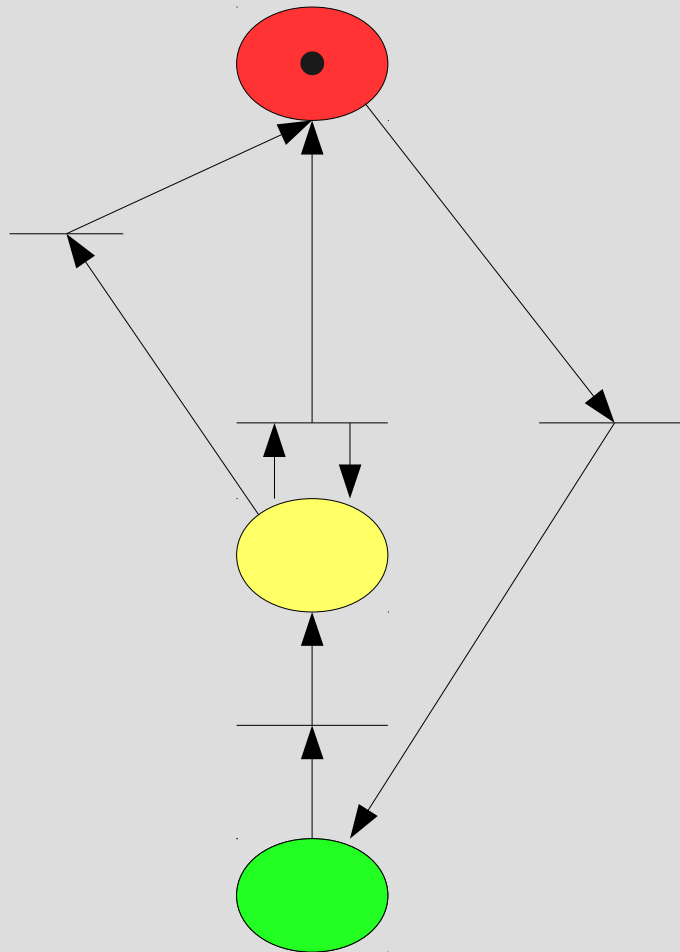
Modelowanie bardziej złożonych sytuacji

- Jak oddać właściwy porządek zmiany świateł?



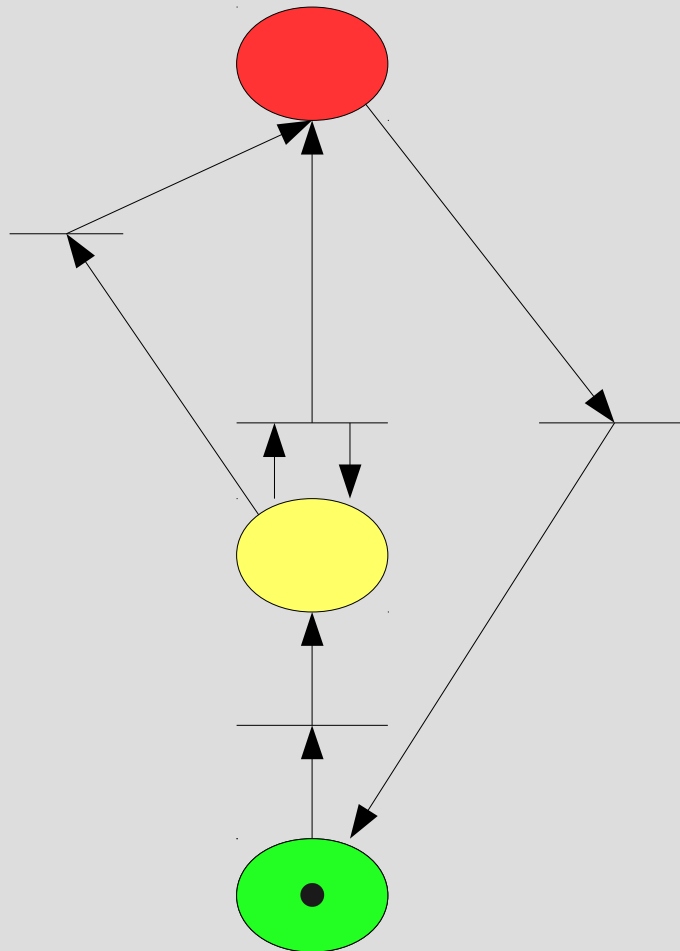
Modelowanie bardziej złożonych sytuacji

- Jak oddać właściwy porządek zmiany świateł?



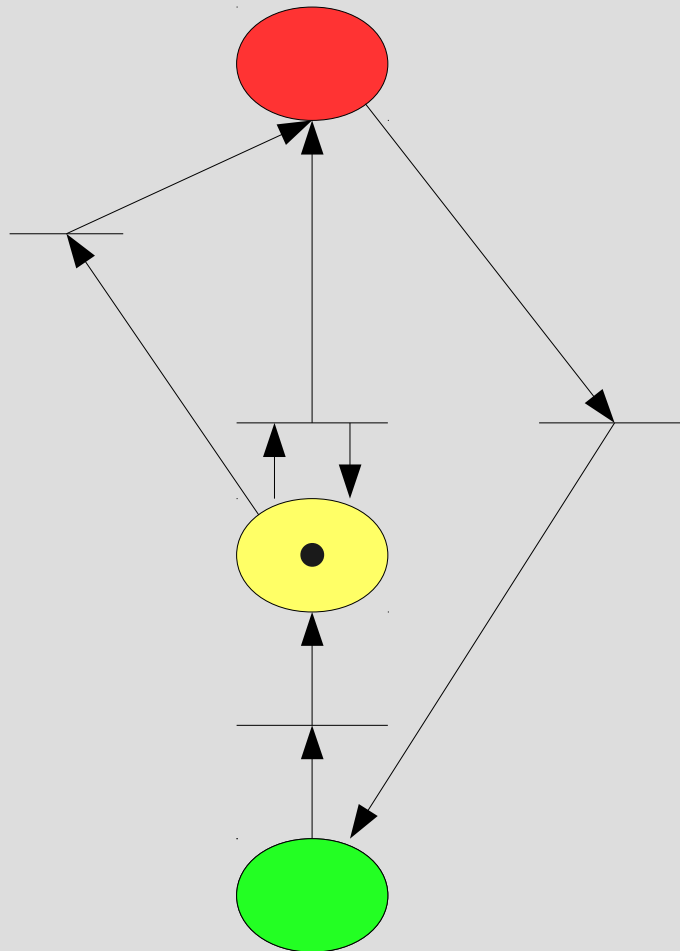
Modelowanie bardziej złożonych sytuacji

- Jak oddać właściwy porządek zmiany świateł?



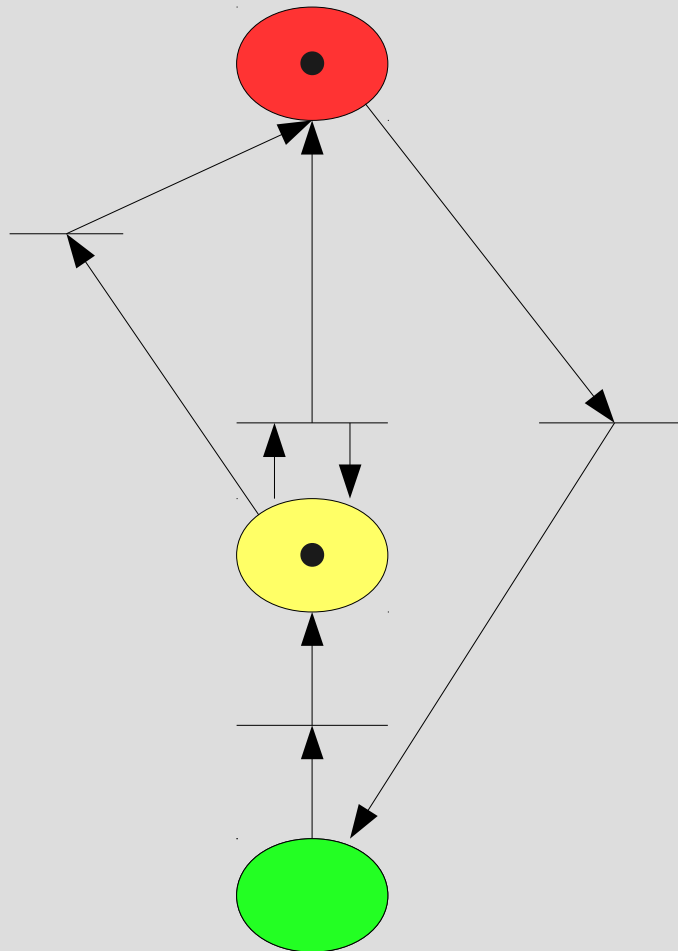
Modelowanie bardziej złożonych sytuacji

- Jak oddać właściwy porządek zmiany świateł?



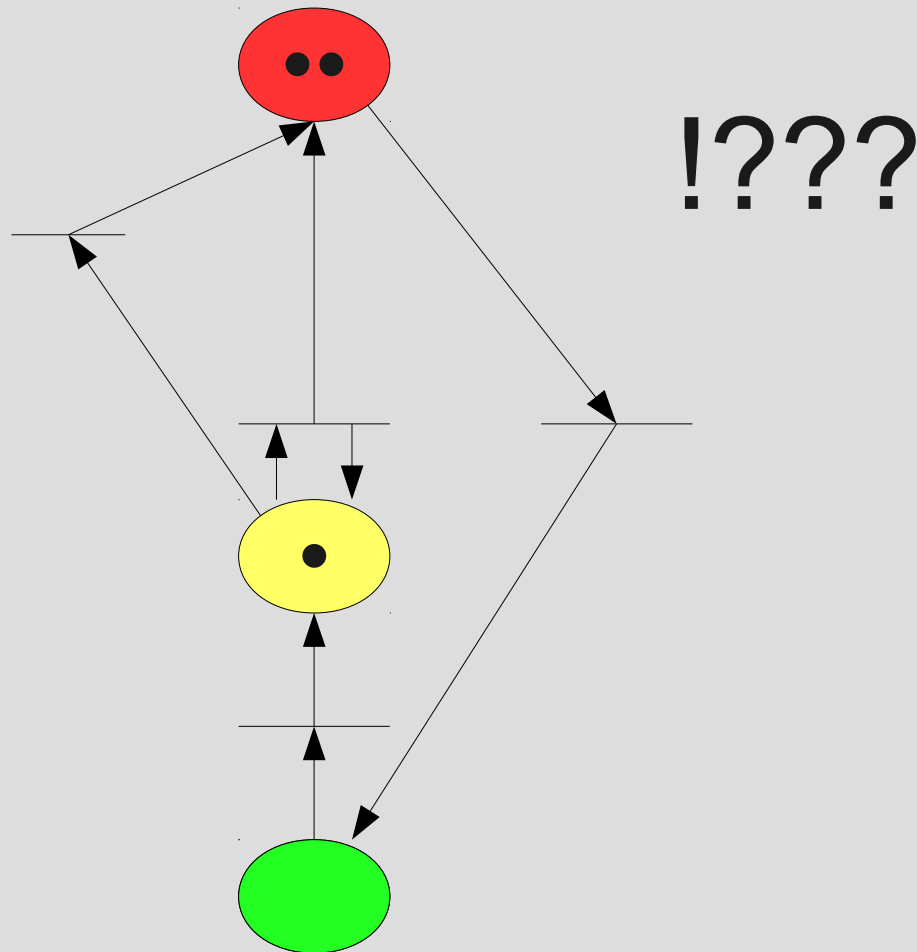
Modelowanie bardziej złożonych sytuacji

- Jak oddać właściwy porządek zmiany świateł?



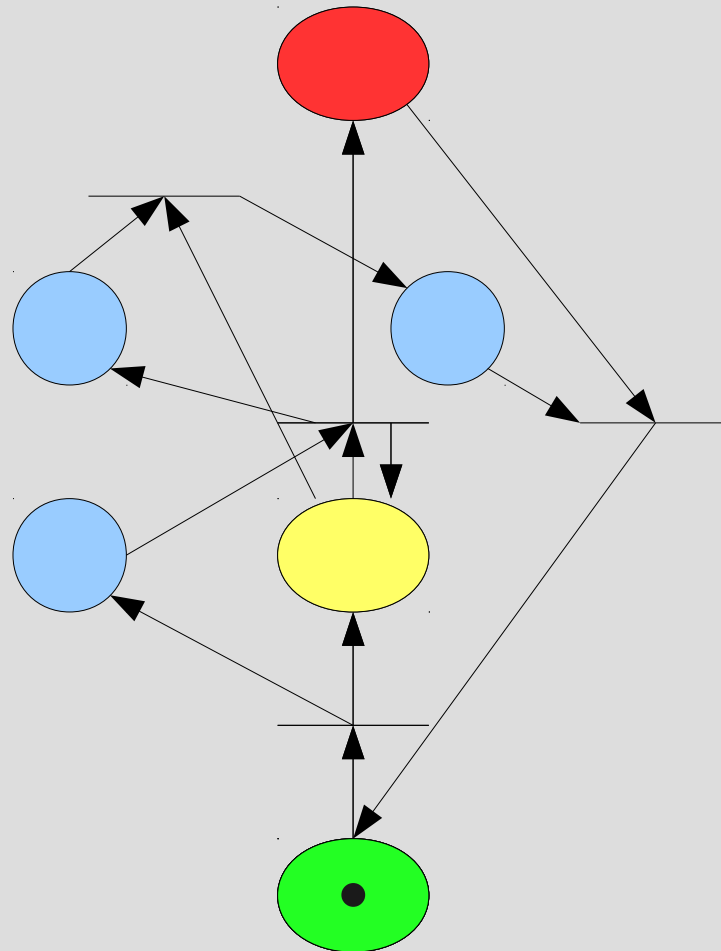
Modelowanie bardziej złożonych sytuacji

- OOPS!



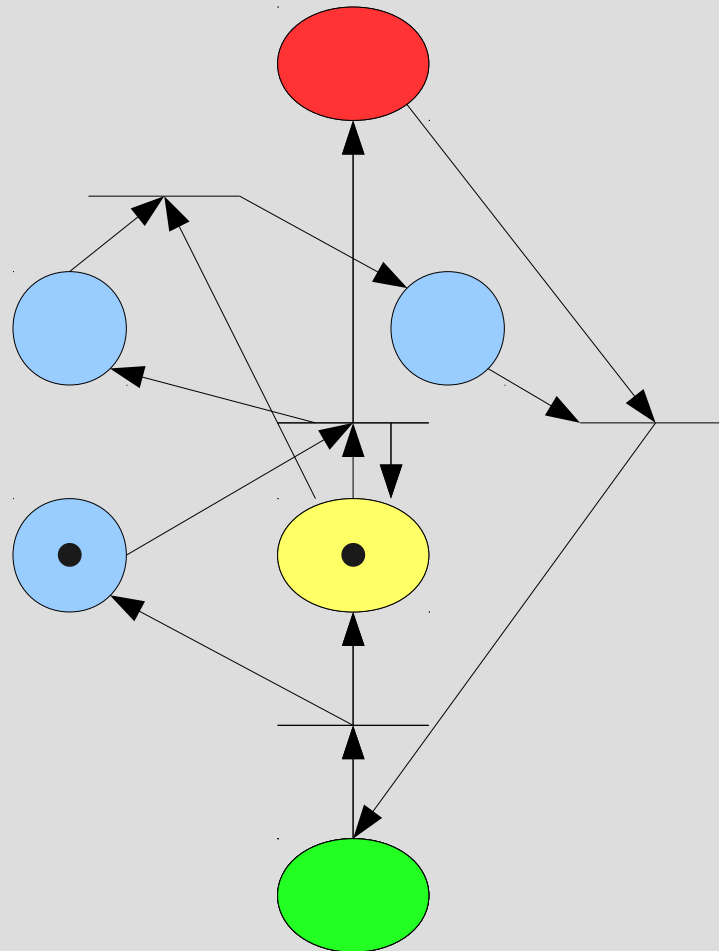
Modelowanie bardziej złożonych sytuacji

- Poprawny model zmiany światła



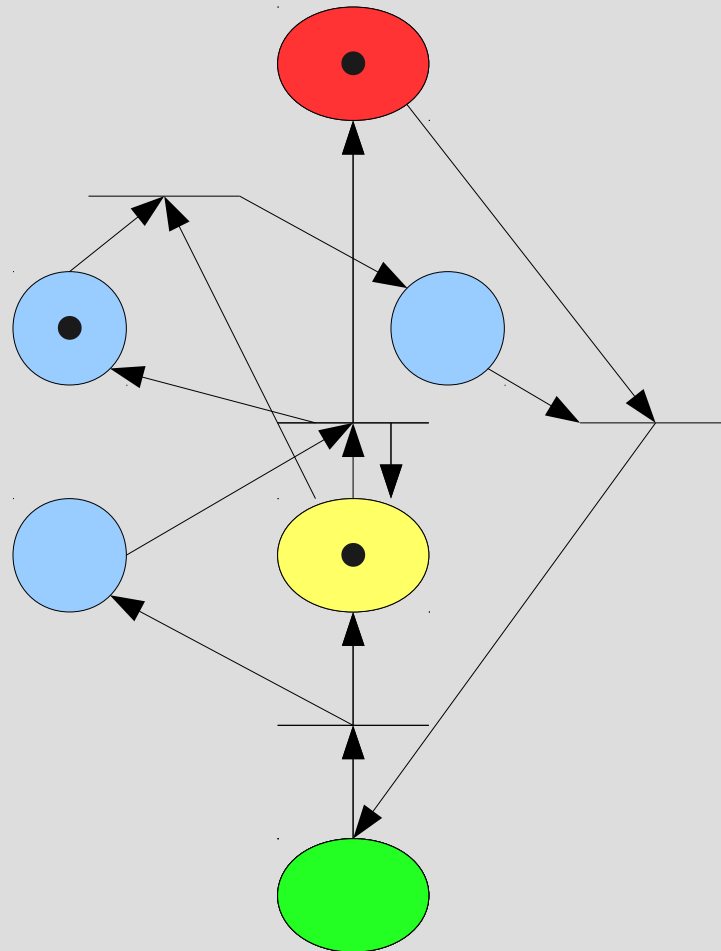
Modelowanie bardziej złożonych sytuacji

- Poprawny model zmiany światła



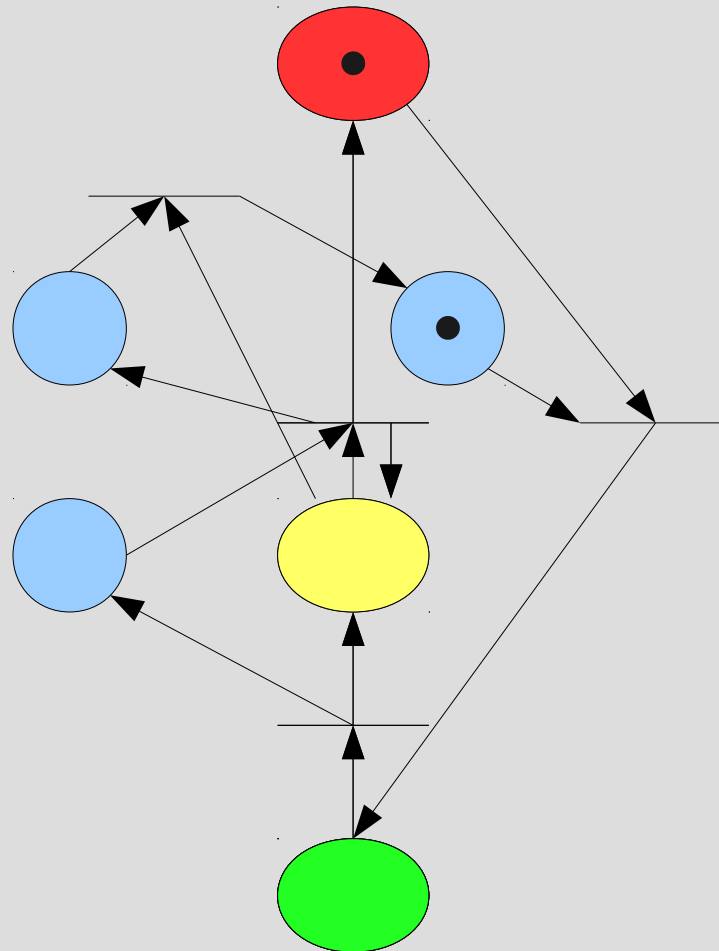
Modelowanie bardziej złożonych sytuacji

- Poprawny model zmiany światła



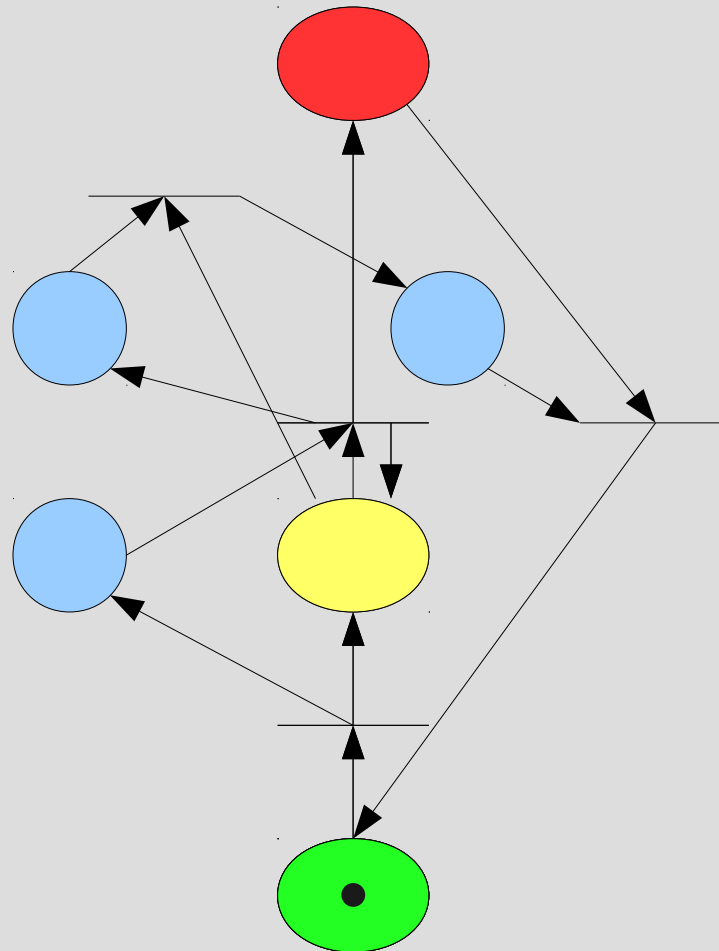
Modelowanie bardziej złożonych sytuacji

- Poprawny model zmiany światła



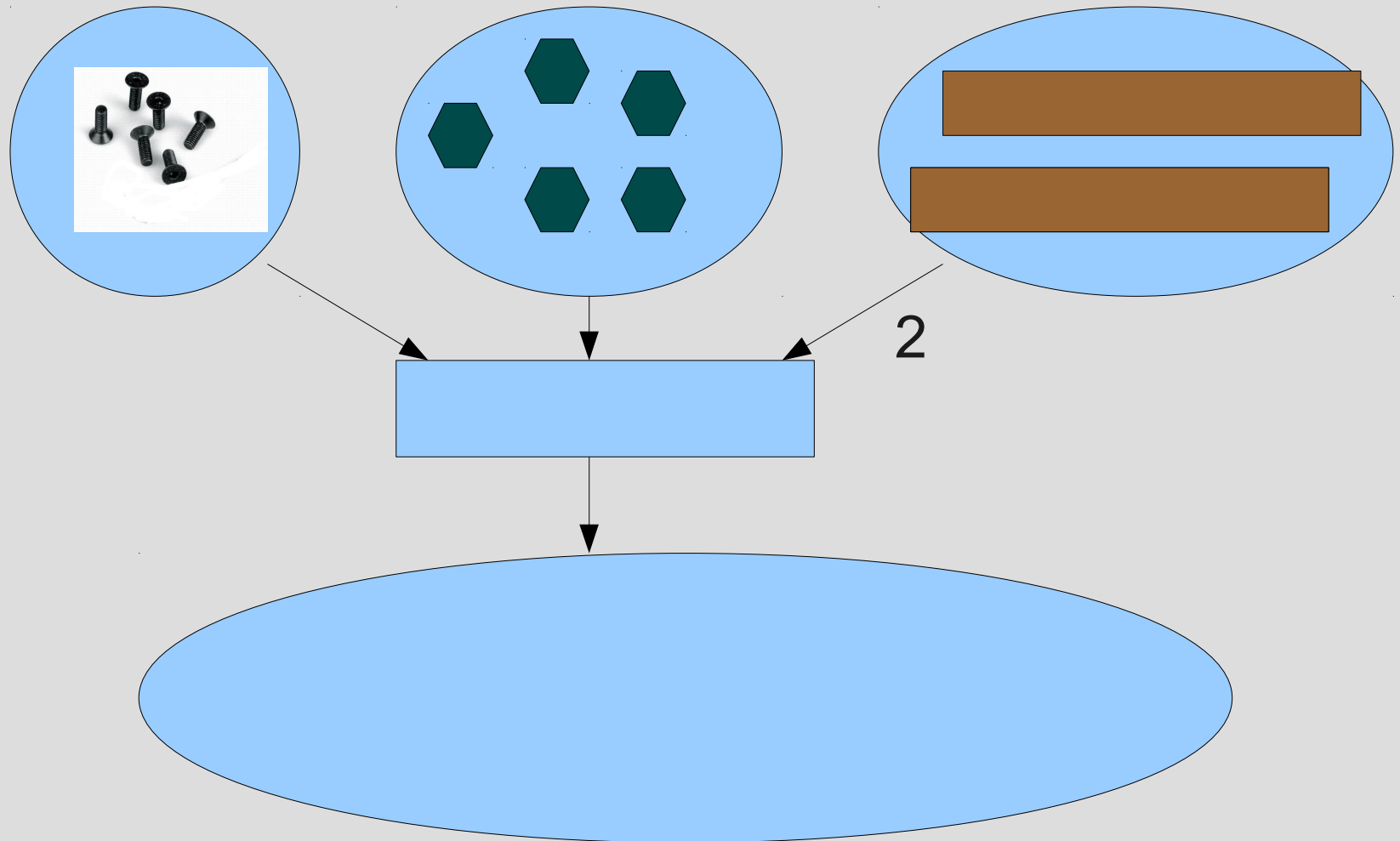
Modelowanie bardziej złożonych sytuacji

- Poprawny model zmiany światła



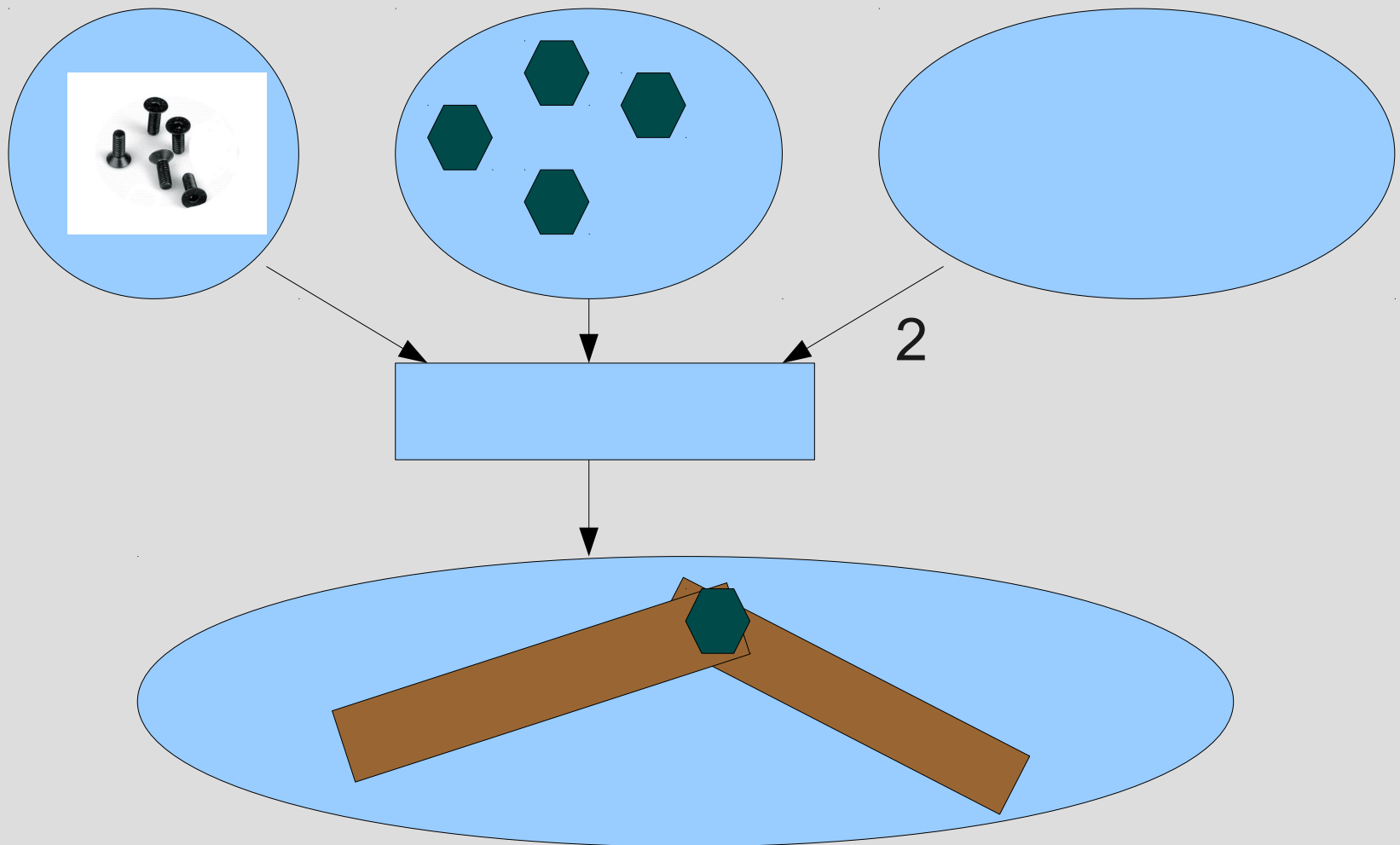
Modelowanie procesów produkcyjnych

- Skręcamy listewki



Modelowanie procesów produkcyjnych

- Skręcamy listewki



Problemy z sieciami wielopionkowymi

- Bardzo się komplikuje analiza stanów osiągalnych z pewnej sytuacji początkowej
- Na niektóre pytania nie da się odpowiedzieć (np. czy dwie dane sieci generują dokładnie takie same przestrzenie stanów).
- Odpowiedzi na te pytania, dotyczące osiągalnych stanów, na które da się odpowiedzieć, mają często bardzo dużą złożoność - wymagają ogromnych zasobów pamięciowych i czasowych, czasem tak dużych, że praktycznie okazują się nierozwiązywalne!

Żywotność i ograniczoność

- Dwie własności przyciągały uwagę teoretyków sieci Petriego od zawsze: żywotność i ograniczoność.
- Dotyczyły one tego, co się może stać, gdy zaprojektujemy sieć i wybierzemy jakiś początkowy układ pionków (markowanie).

Żywotność

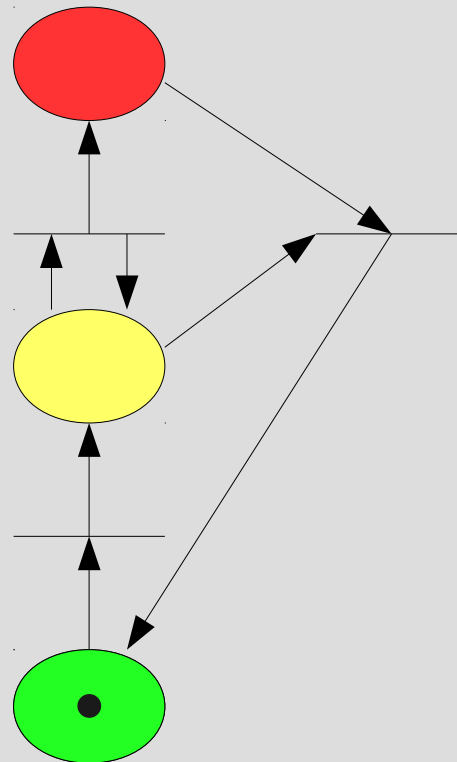
- Powiemy, że markowanie w sieci jest żywe, jeśli nigdy w przyszłości (przy odpalaniu zdarzeń z tego markowania) nie może się zdarzyć, aby któreś zdarzenie zostało pozbawione możliwości zajścia.
- Żywotność jest typowa dla procesów cyklicznych.

Ograniczoność

- Markowanie w sieci jest ograniczone, jeśli w żadnym z jego miejsc w trakcie działania sieci liczba pionków nie może rosnać w nieskończoność.
- W przypadku sieci z ograniczonymi markowaniami możemy być pewni, że liczba stanów osiągalnych jest skończona, choć może być bardzo duża (nawet dla niewielkich sieci większa niż liczba w atomów we Wszechświecie).

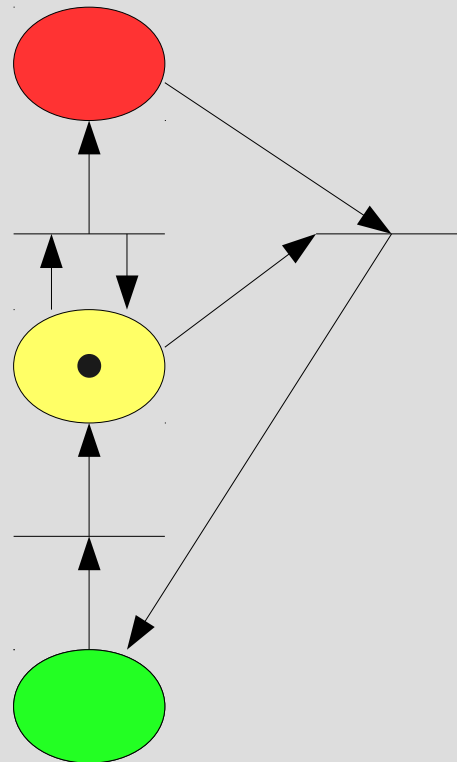
Przykład markowania nieograniczonego

- Tutaj w czerwonym miejscu (ale tylko w tym!) możemy napompować dowolnie dużo pionków



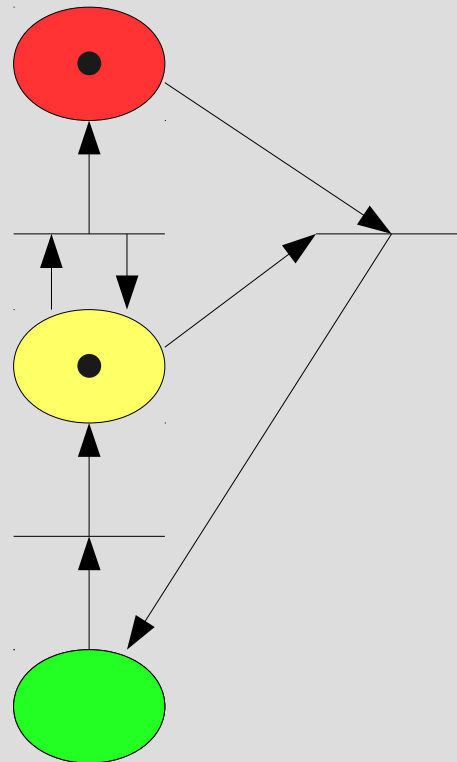
Przykład markowania nieograniczonego

- Tutaj w czerwonym miejscu (ale tylko w tym!) możemy napompować dowolnie dużo pionków



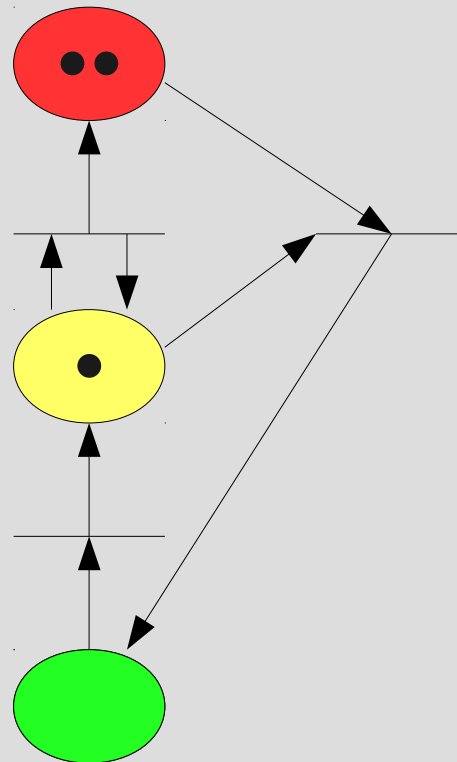
Przykład markowania nieograniczonego

- Tutaj w czerwonym miejscu (ale tylko w tym!) możemy napompować dowolnie dużo pionków



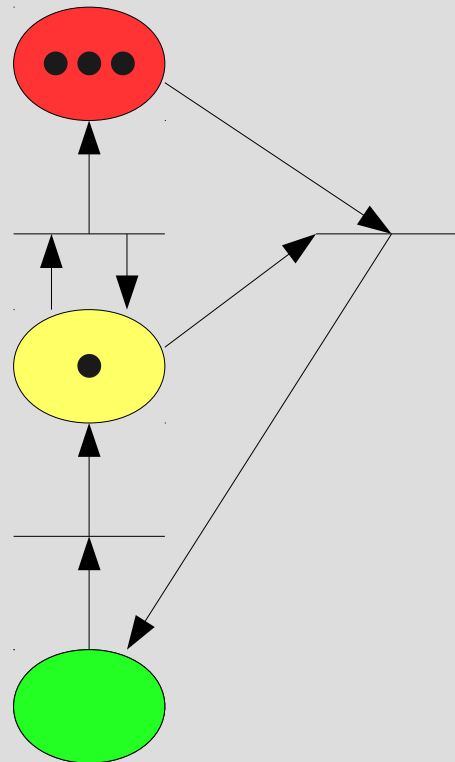
Przykład markowania nieograniczonego

- Tutaj w czerwonym miejscu (ale tylko w tym!) możemy napompować dowolnie dużo pionków



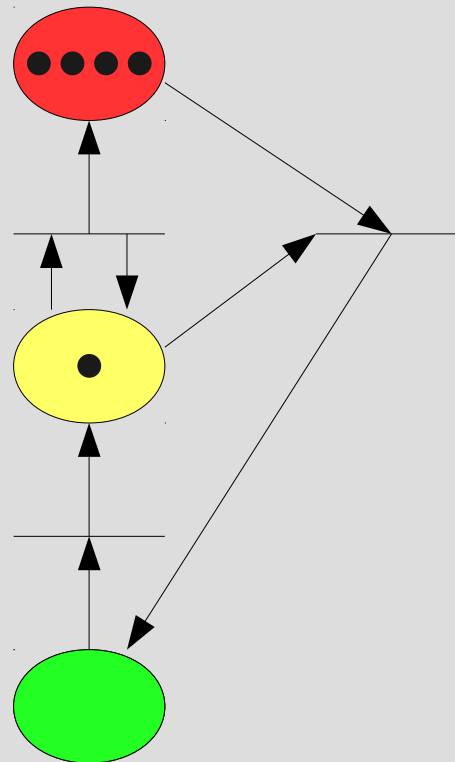
Przykład markowania nieograniczonego

- Tutaj w czerwonym miejscu (ale tylko w tym!) możemy napompować dowolnie dużo pionków



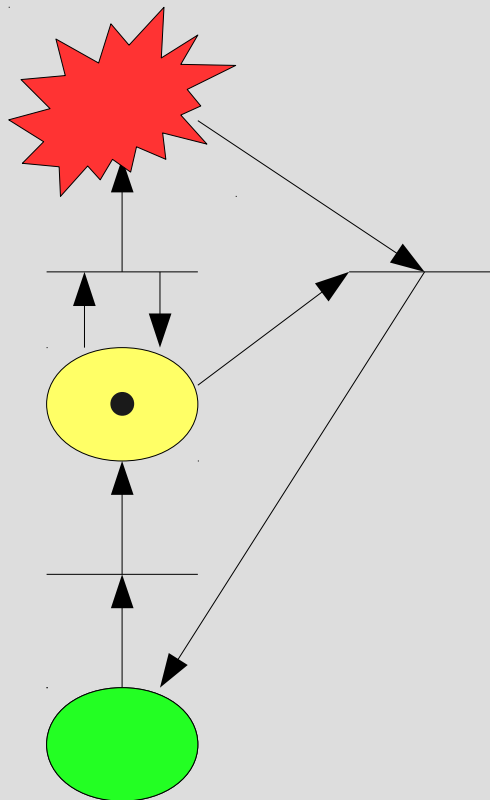
Przykład markowania nieograniczonego

- Tutaj w czerwonym miejscu (ale tylko w tym!) możemy napompować dowolnie dużo pionków



Przykład markowania nieograniczonego

- Tutaj w czerwonym miejscu (ale tylko w tym!) możemy napompować dowolnie dużo pionków



Sieci dobrze zbudowane

- Sieć jest dobrze zbudowana, jeśli istnieje w niej markowanie żywe i ograniczone.
- Z jednej strony chodzi o to, żeby w sieci, która ma działać cyklicznie, nie dochodziło do wyłączeń po wsze czasy jakichkolwiek jej części
- Z drugiej chodzi o to, żeby nie doszło do przepełnienia buforów na dane (w miejscach sieci często odwzorowujemy ilościowo jakieś dane).

Jak to się ma do przepływów pracy?

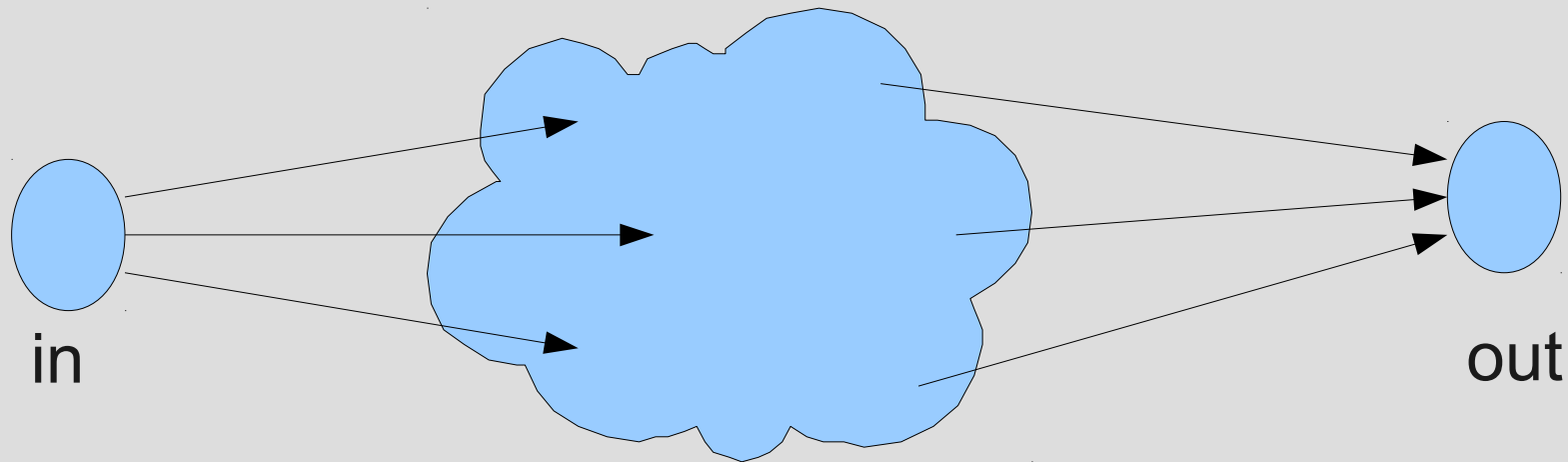
- Według Wikipedii:
 - Workflow (przepływ pracy), to pojęcie określające *sposób przepływu informacji* pomiędzy różnymi obiektami biorącymi udział w jej przetwarzaniu.
 - W węższym sensie jest to określenie *sposobu przepływu dokumentów* pomiędzy pracownikami wykonującymi pewien zalgorytmizowany zespół czynności.

Workflow wg koalicji WFMC (Workflow Management Coalition)

- Przepływ pracy, to automatyzacja procesów biznesowych, w całości lub w części, podczas której dokumenty, informacje lub zadania są przekazywane od jednego uczestnika do następnego, według odpowiednich procedur zarządczych.

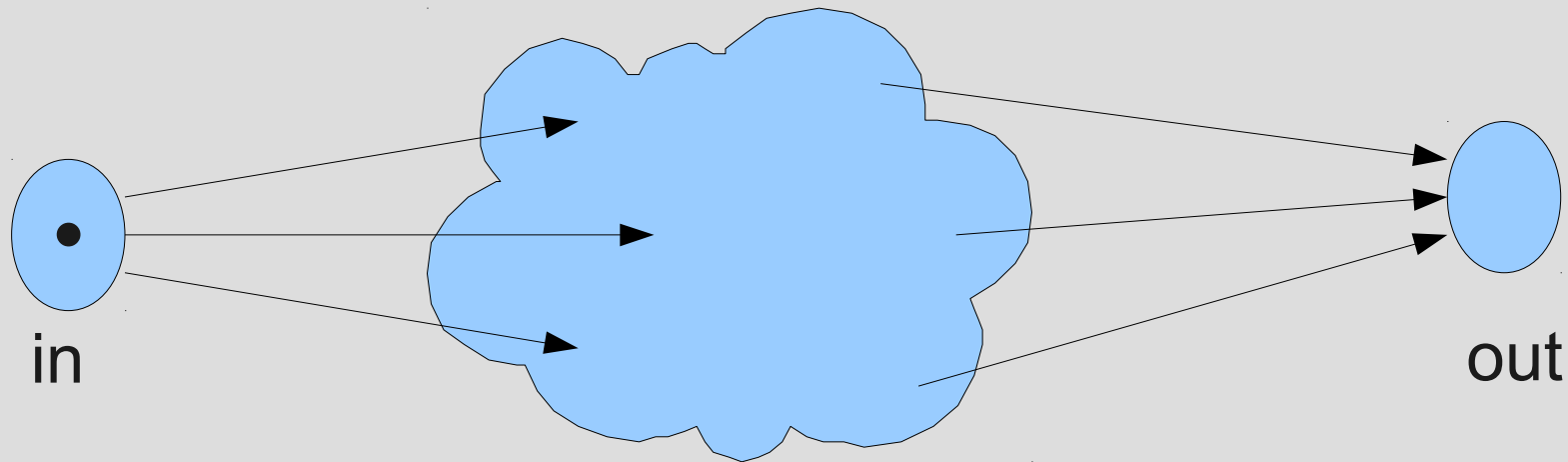
Sieci przepływu pracy

- Ogólny schemat przepływu pracy:



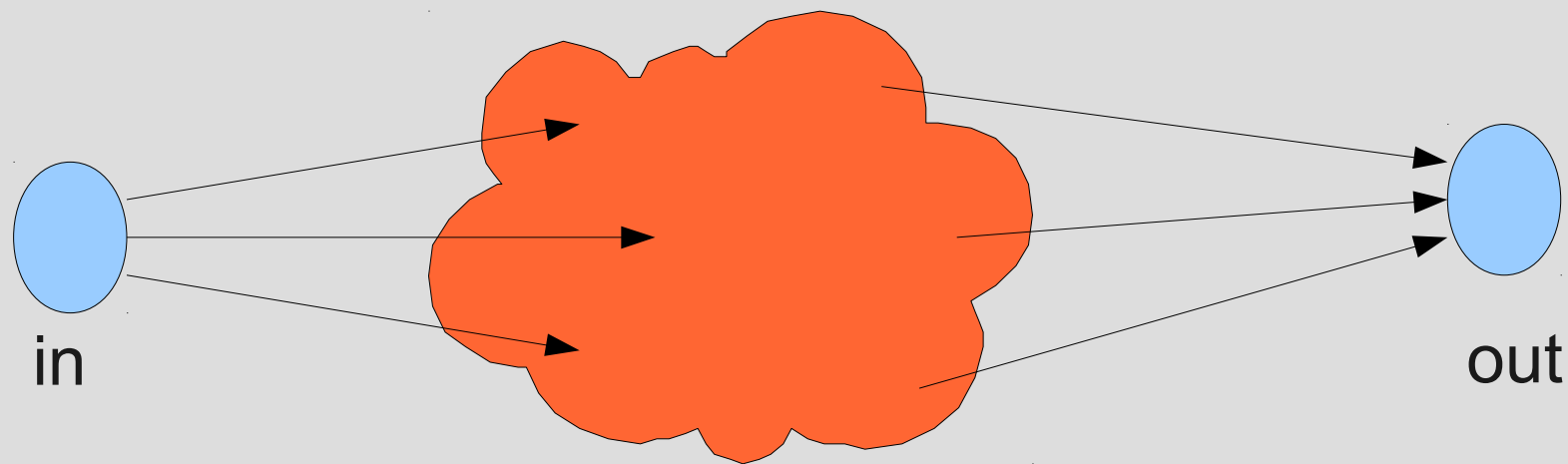
Sieci przepływu pracy

- Ogólny schemat przepływu pracy:



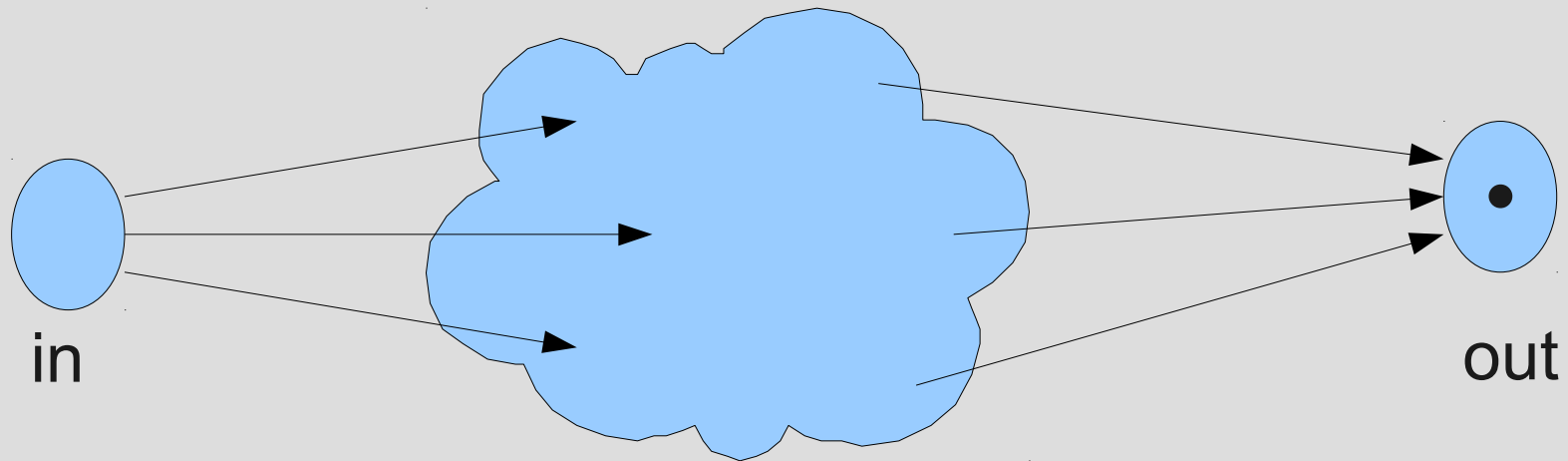
Sieci przepływu pracy

- Ogólny schemat przepływu pracy:



Sieci przepływu pracy

- Ogólny schemat przepływu pracy:



WF-sieć

- Sieć przepływu pracy (WF-sieć), to sieć Petriego z dwoma wyróżnionymi miejscami in, out taka, że:
 - nie ma strzałek wchodzących do in, ani wychodzących z out
 - każdy węzeł leży na ścieżce od in do out
 - początkowe markowanie zawiera pionek w miejscu in i nigdzie więcej.

Poprawne (sound) WF-sieci

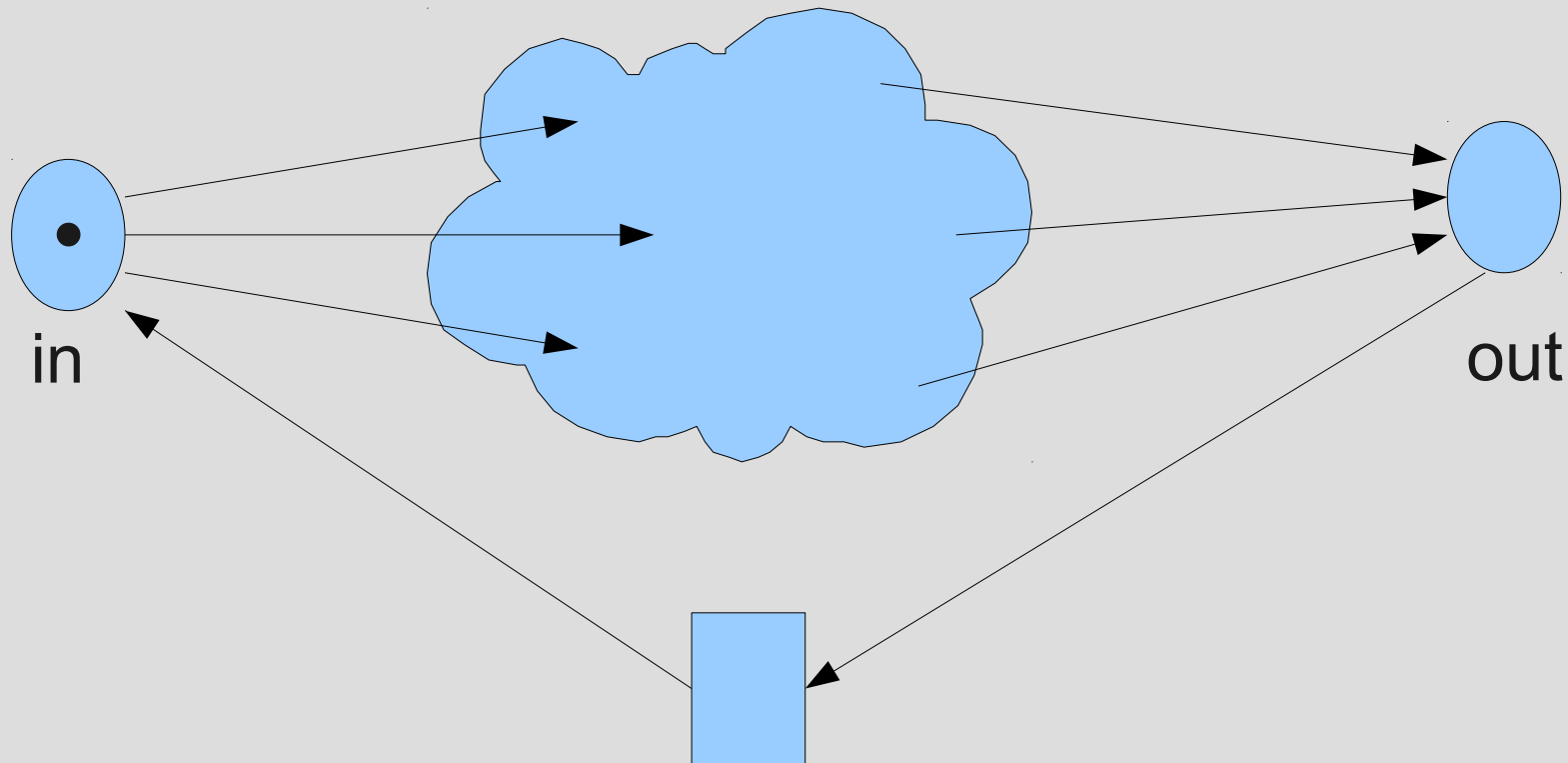
- Powiemy, że WF-sieć jest poprawna, jeśli:
 - Każde zdarzenie ma szansę zajść jeśli zaczniemy od początkowego markowania
 - Przy dowolnym ciągu odpaleń z markowania początkowego zawsze można dojść do markowania końcowego zawierającego jeden pionek w miejscu out i nigdzie więcej.
- Sieci poprawne nie mają zbędnych tranzycji i
 - co można pokazać - przy dojściu do stanu końcowego nie pozostawiają „śmieci”, czyli zagubionych pionków wewnątrz „chmurki”. Kończą „na czysto”.

Sieci strukturalne, a poprawność

- Reguły rozdrobnień zapewniają poprawność powstającej sieci.
- Jedyne niebezpieczeństwa mogą kryć się za sklejeniem miejsc zasobowych - ten problem wymaga większej ostrożności i jest tematem powstającej pracy doktorskiej jednego z moich studentów.

Domknięta WF-sieć

- Zacyklając sieć przepływu pracy dostajemy domkniętą WF-sieć.



Twierdzenie van der Aalsta

- WF sieć jest poprawna wtedy i tylko wtedy gdy jej początkowe markowanie jest żywe i ograniczone (czyli sieć jest dobrze zbudowana).

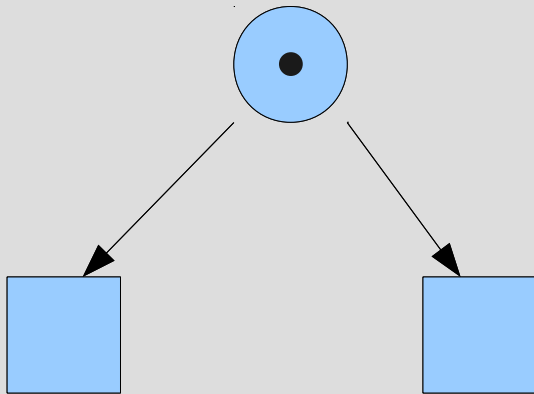
Ciekawe, że pojęcia, które teoretycy
od lat badali
okazały się istotne w praktyce!

Dygresja - twierdzenie o rzędzie

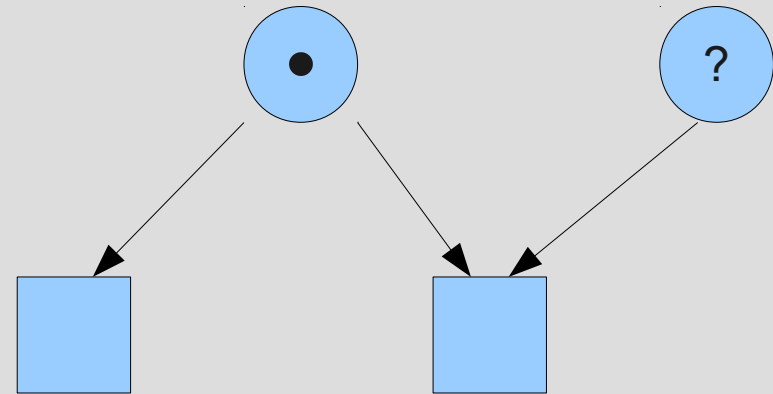
- Przy okazji własność dobrego zbudowania sieci (Żywotność + ograniczoność) jest przepięknie scharakteryzowana przez słynne twierdzenie o rzędzie. Dotyczy ona *sieci Petriego z wolnych wyborem*, czyli takich, że gdy gdziekolwiek dwa zdarzenia walczą o zasoby jednego miejsca, to to miejsce jest jedynym ich miejscem wejściowym.

Sieci Petriego z wolnym wyborem

- Wolny wybór, to nieoglądanie się na inne warunki, gdy dochodzi do konfliktu



Wolny wybór



Nie wolny wybór

Macierz incydencji

- Numerujemy zdarzenia i miejsca w sieci.
- Dla sieci Petriego tworzymy macierz C , która ma tyle wierszy, ile jest miejsc i tyle kolumn ile jest tranzycji.
- Na przecięciu i -tego wiersza i j -tej kolumny jest 1, jeśli j -te zdarzenie dostarcza pionek i -temu miejscu, -1, jeśli zabiera i 0 jeśli nie jest z nim połączone żadną strzałką.

Twierdzenie o rzędzie

- Jeśli miejsc jest n , tranzycji m , a strzałek prowadzących od miejsc do zdarzeń jest a , to sieć Petriego z wolnym wyborem jest dobrze zbudowana wtedy i tylko wtedy (sic!), gdy są spełnione następujące trzy warunki:

- Istnieje dodatni wektor X taki, że $CX=0$
- Istnieje dodatni wektor Y taki, że $Y^T C=0$
- $\text{rank}(C)=m+n-a-1$

Strukturalne podejście

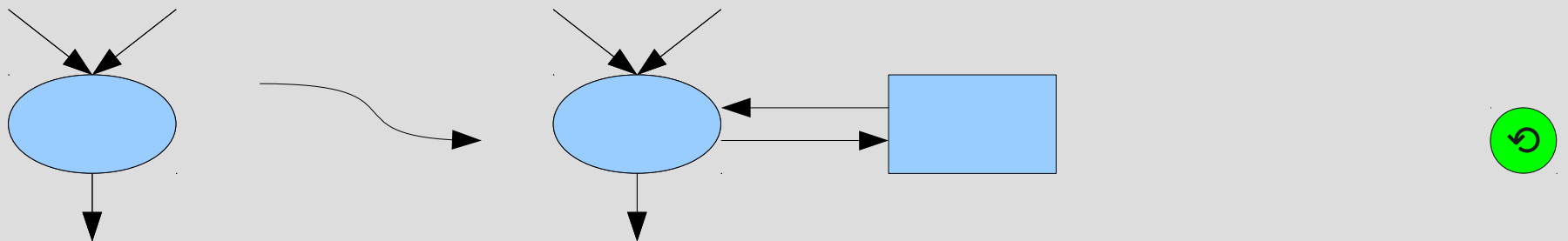
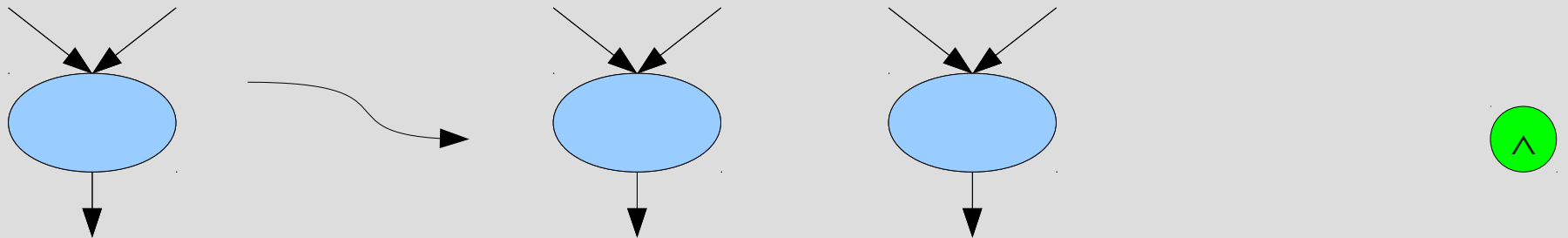
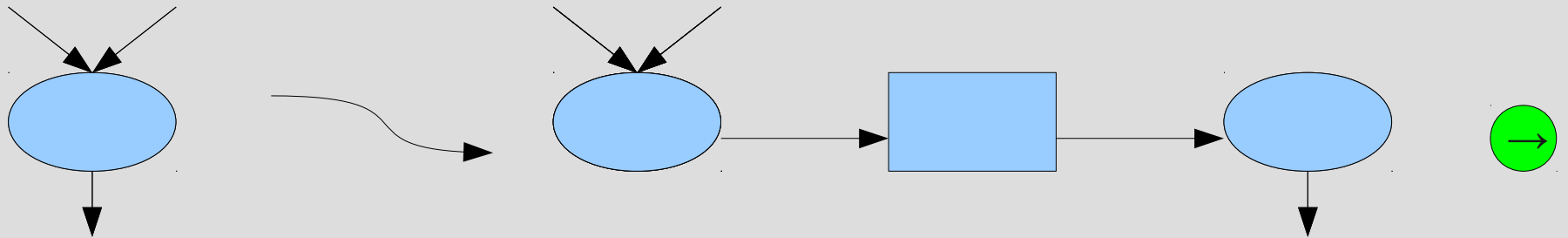
- Jednym z największych przełomów w inżynierii programowania było zdanie sobie sprawy z istoty programowania strukturalnego.
- Dużych programów inaczej po prostu nie da się napisać.
- Wprowadzenie paradygmatu programowania strukturalnego wymagało pewnego wysiłku. Opór był solidny!

Czy można strukturalnie projektować WF-sieci?

- TAK! Trzeba nieco przestawić myślenie metodą zstępującą (top-down)
- Proponuję pewien zestaw podstawowych reguł. Zaczynamy od pojedynczego miejsca, które będziemy stopniowo rozdrabniać stosując jedną z reguł z następujących dwóch slajdów.

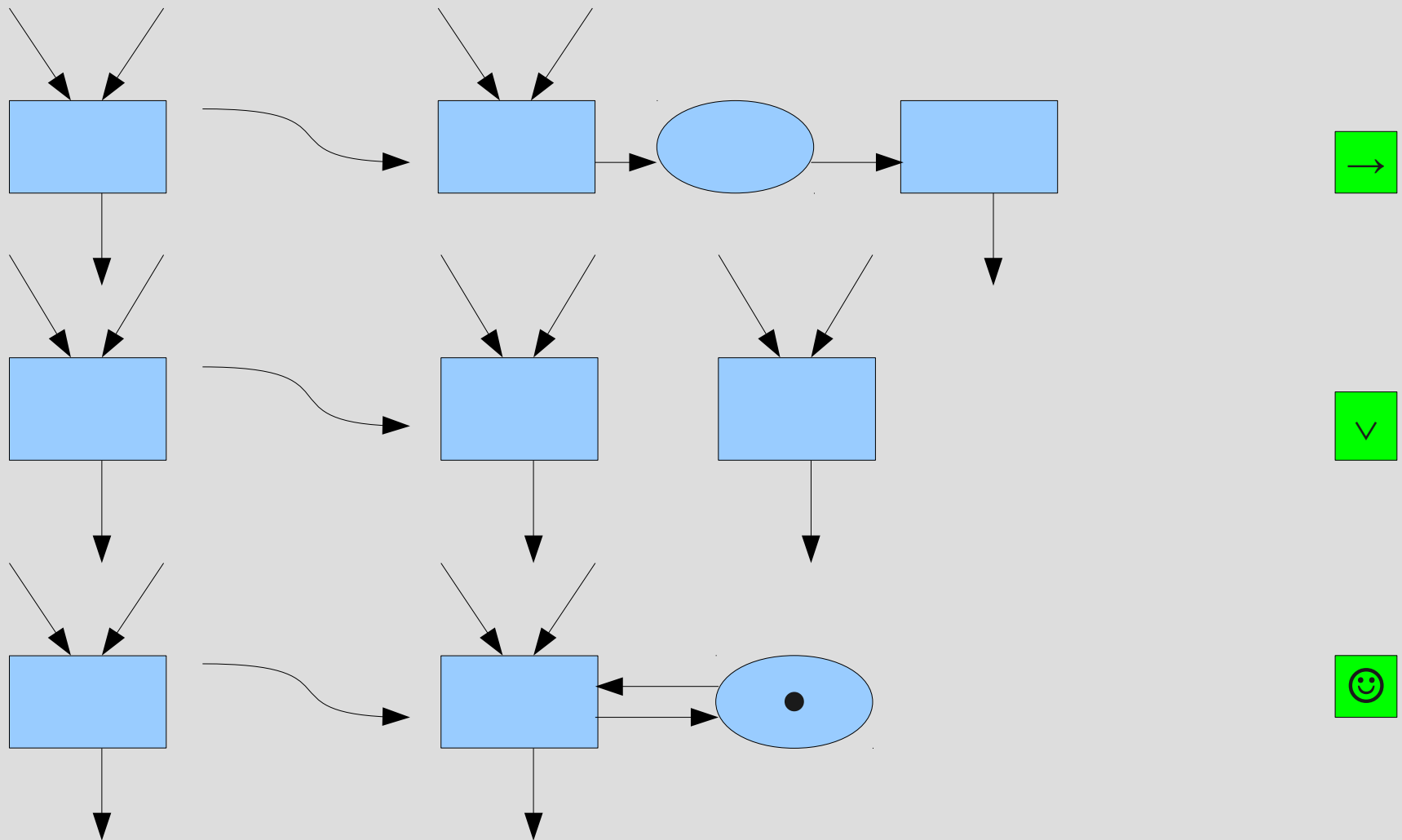
Podstawowe reguły rozdrobnień

- Reguły rozdrobnień dla miejsc



Podstawowe reguły rozdrobnień

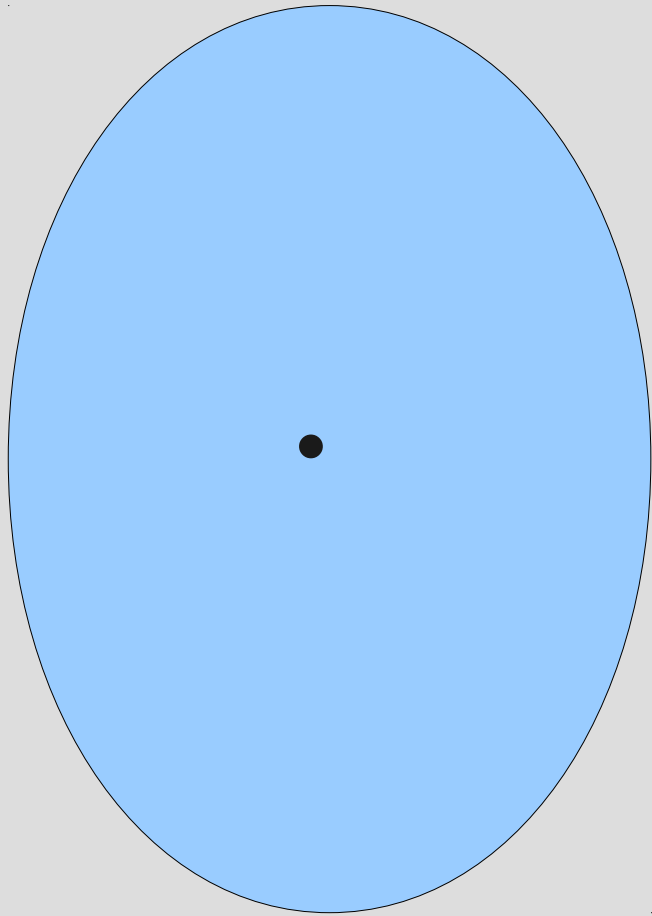
- Reguły rozdrobnień dla tranzycji

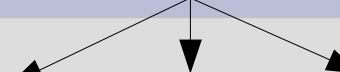
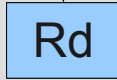
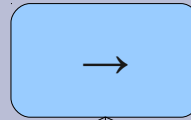
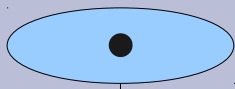


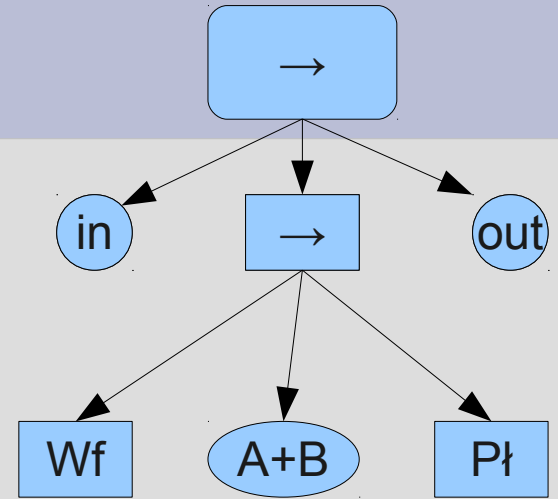
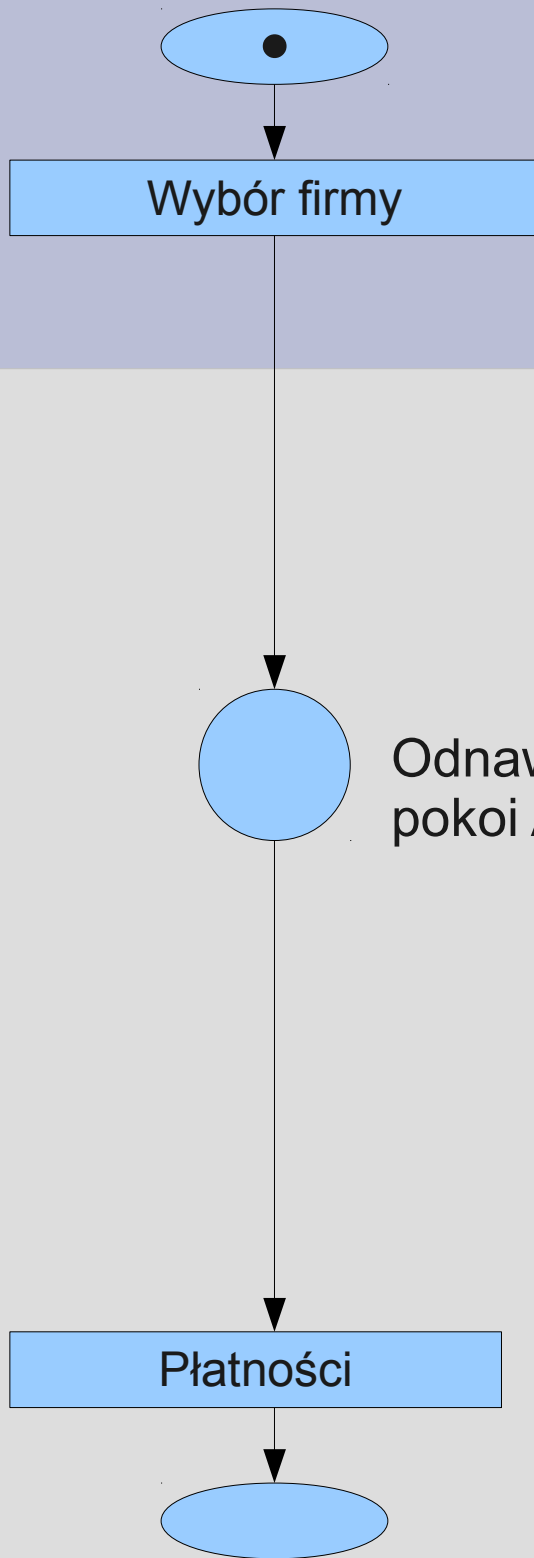
Przykład tynkowania i malowania dwóch pokoi

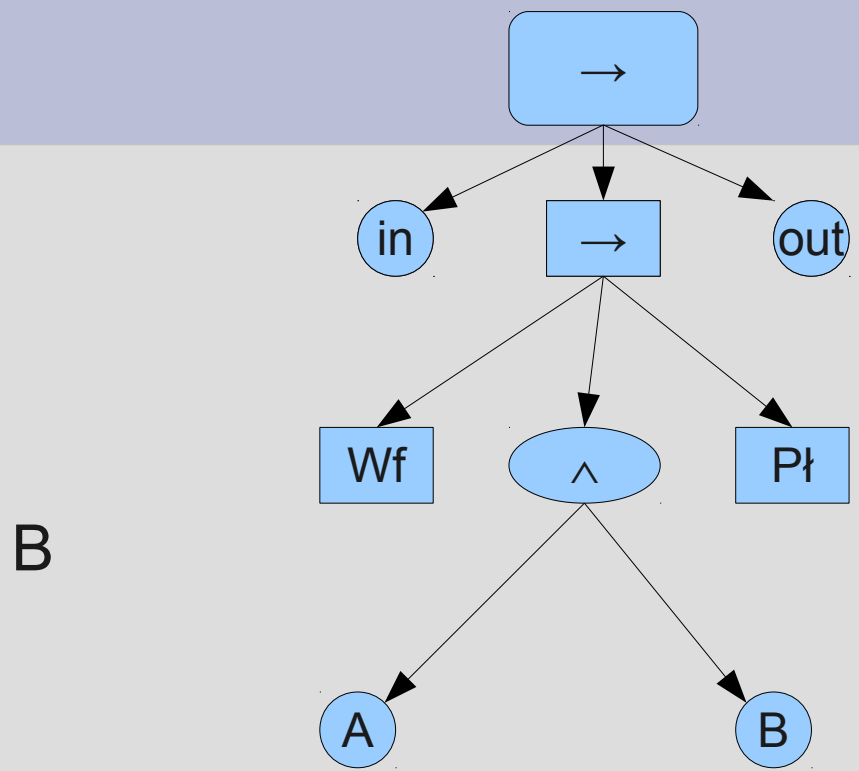
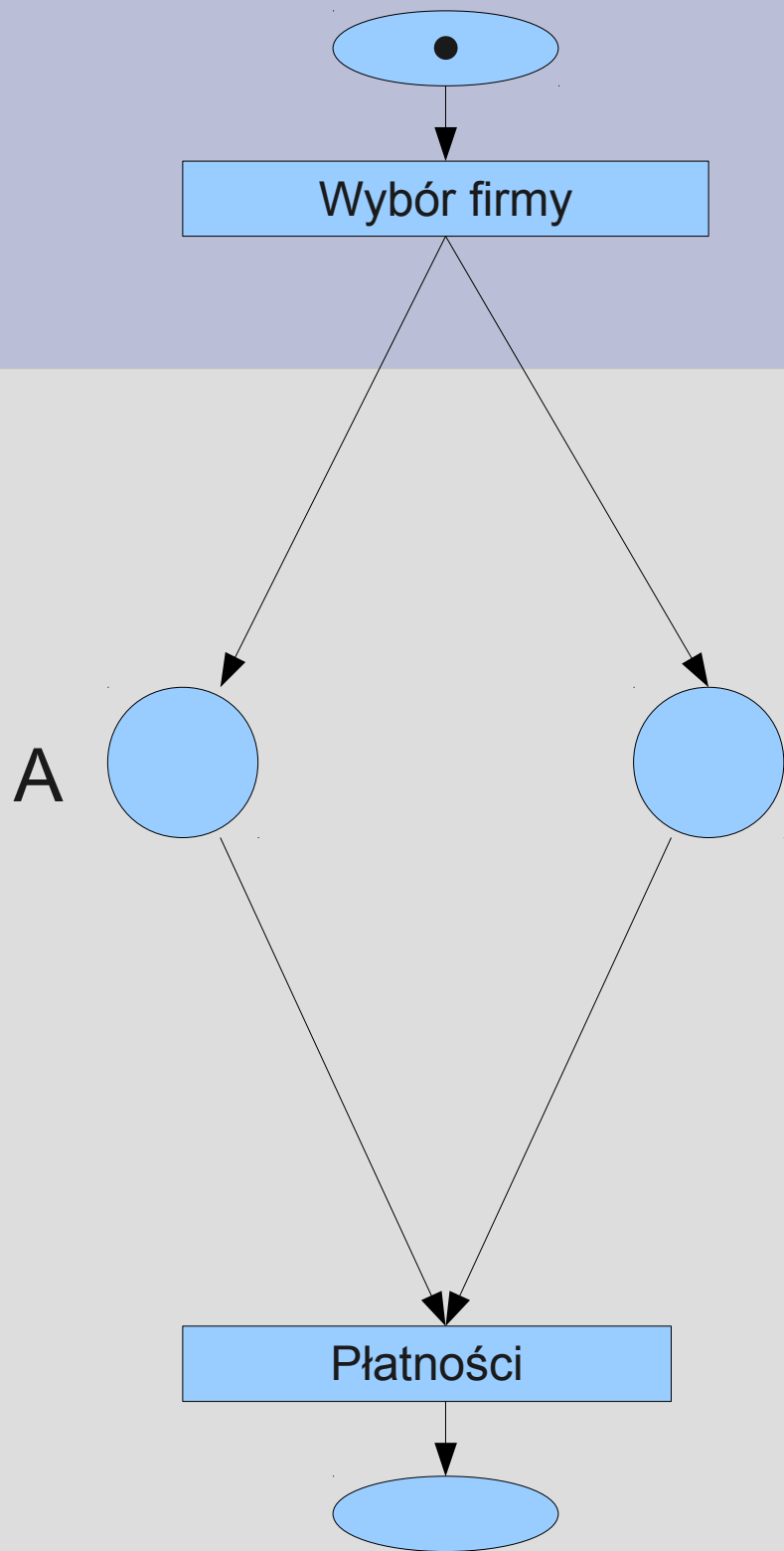
- Pokażemy w etapach, jak można zaprojektować i przeanalizować cały proces.
 - Każdy z pokoi (A i B) wymaga otynkowania i pomalowania
 - Oba tynkowania powinno się zrobić przed malowaniami
 - Jeden z pokoi (A) po pomalowaniu ma być jeszcze udekorowany stiukami i szlaczkiem
 - Do tynkowania potrzeba tynkarza, do malowania malarza, a do dekorowania dekoratora

Remont

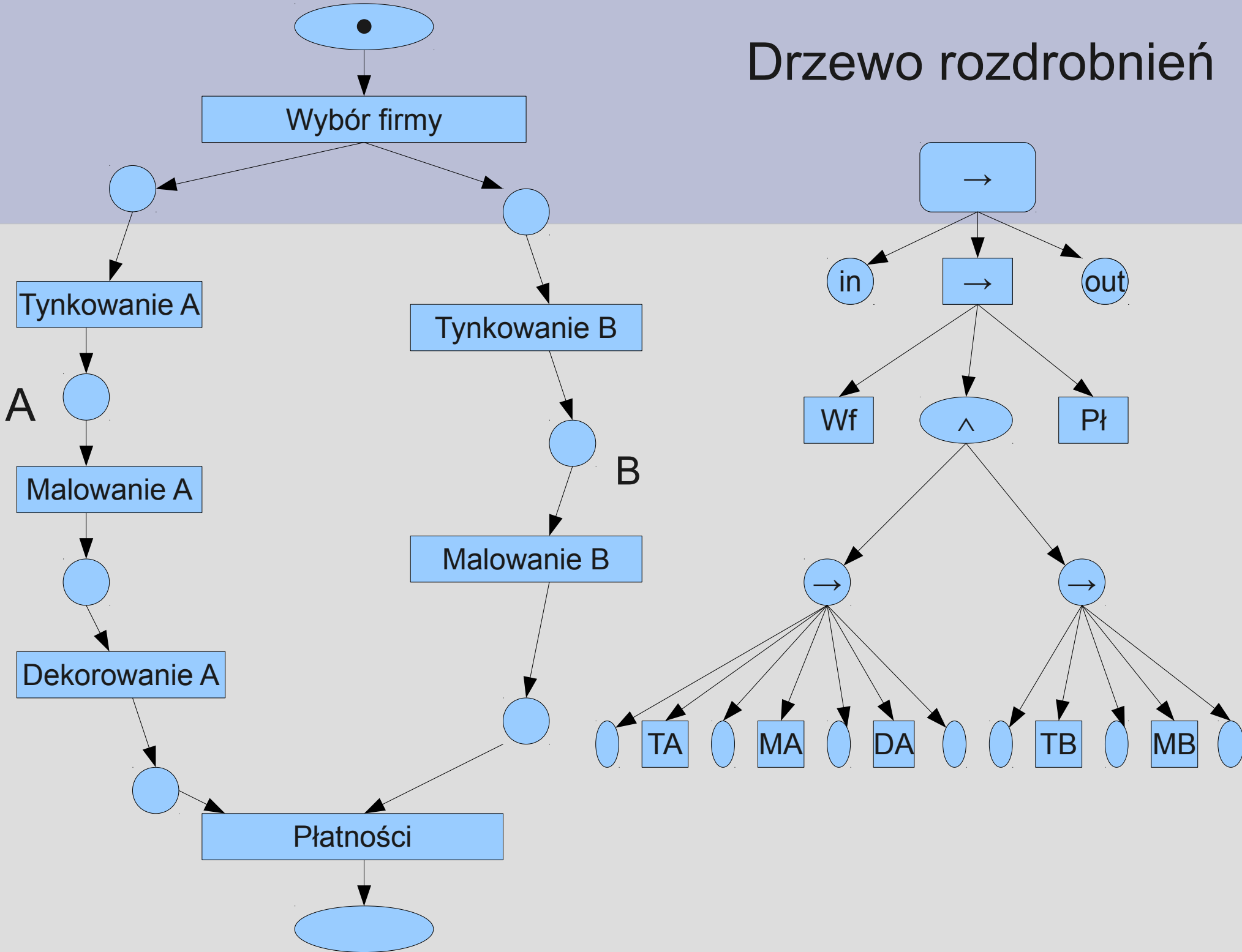


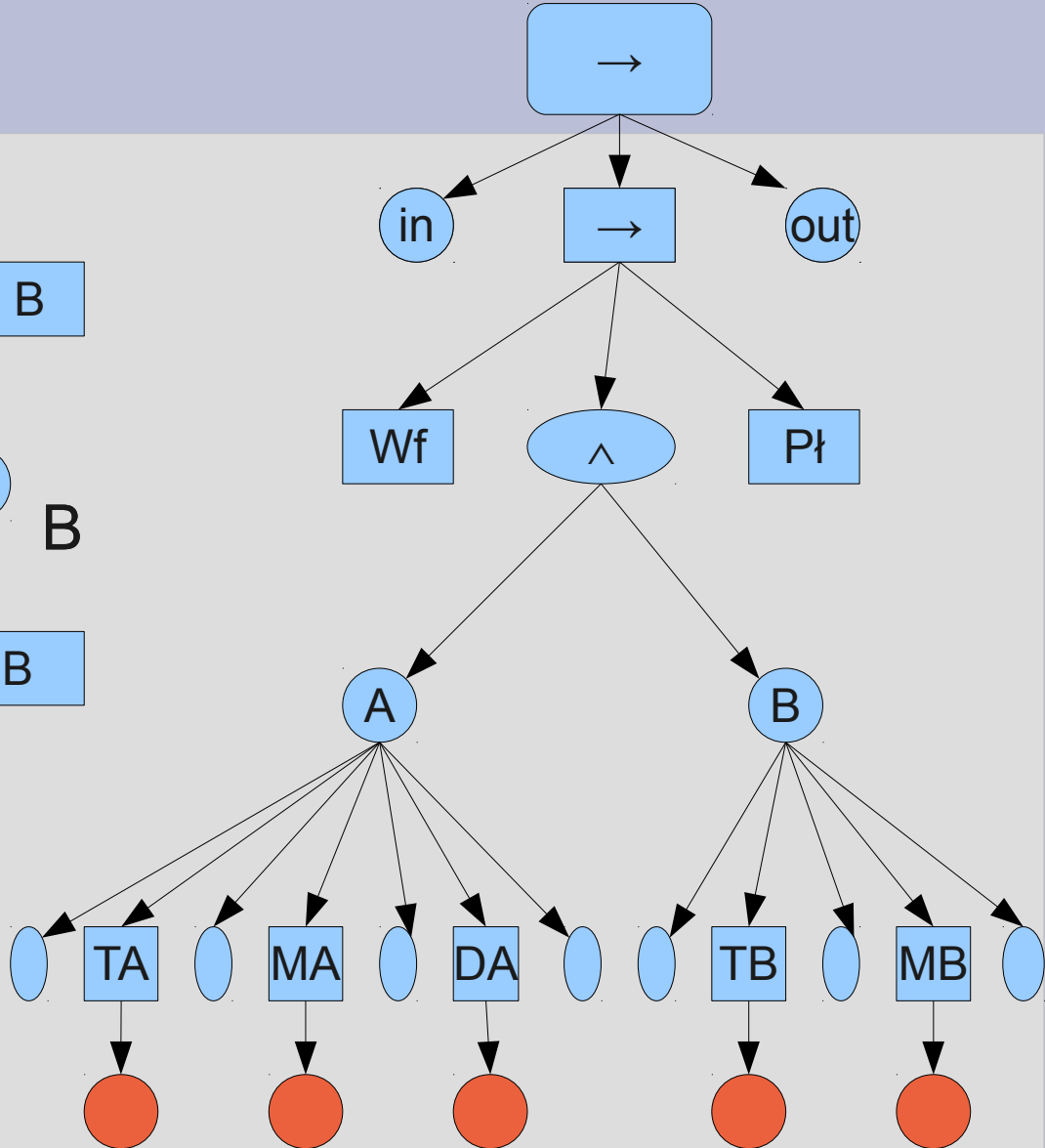
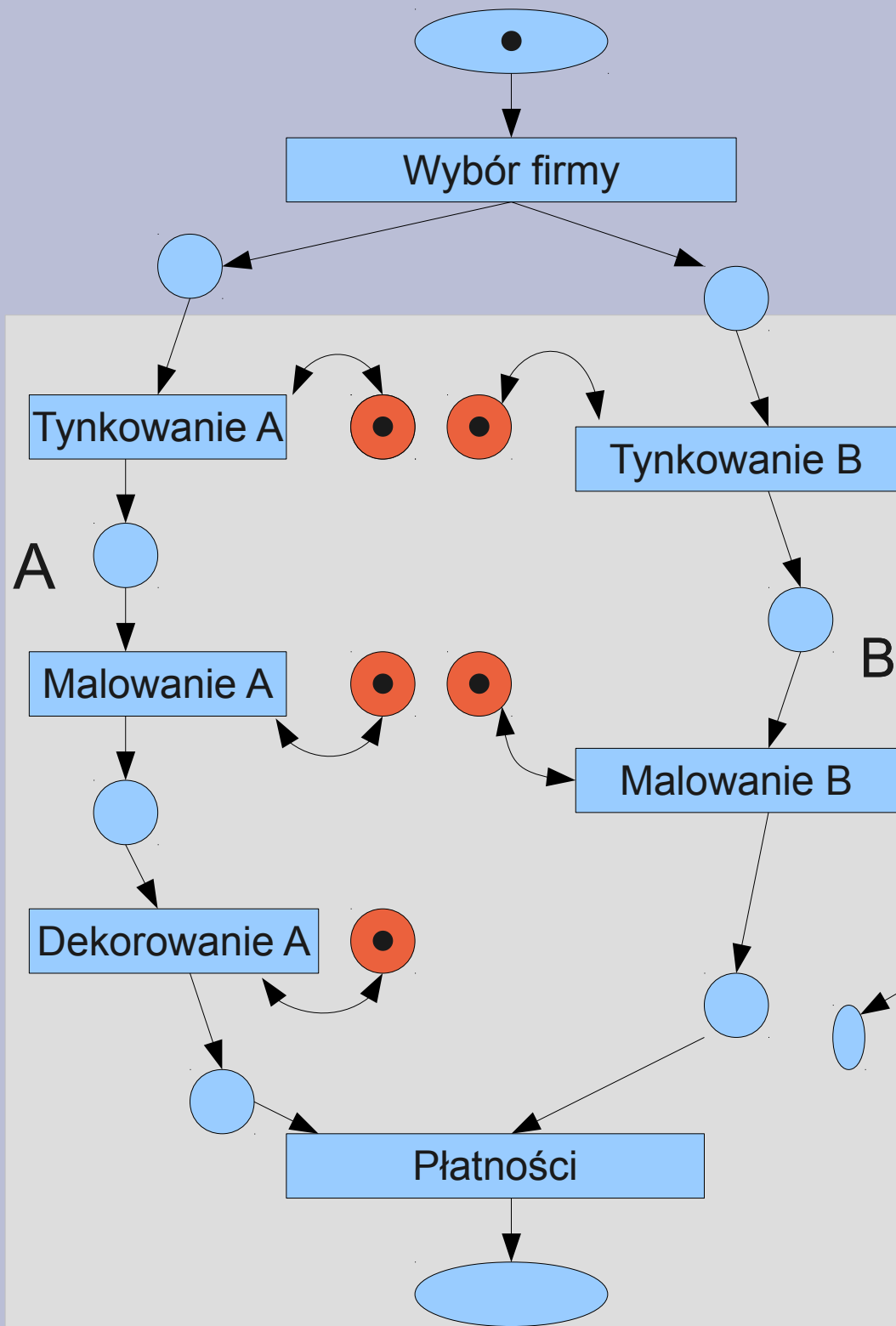


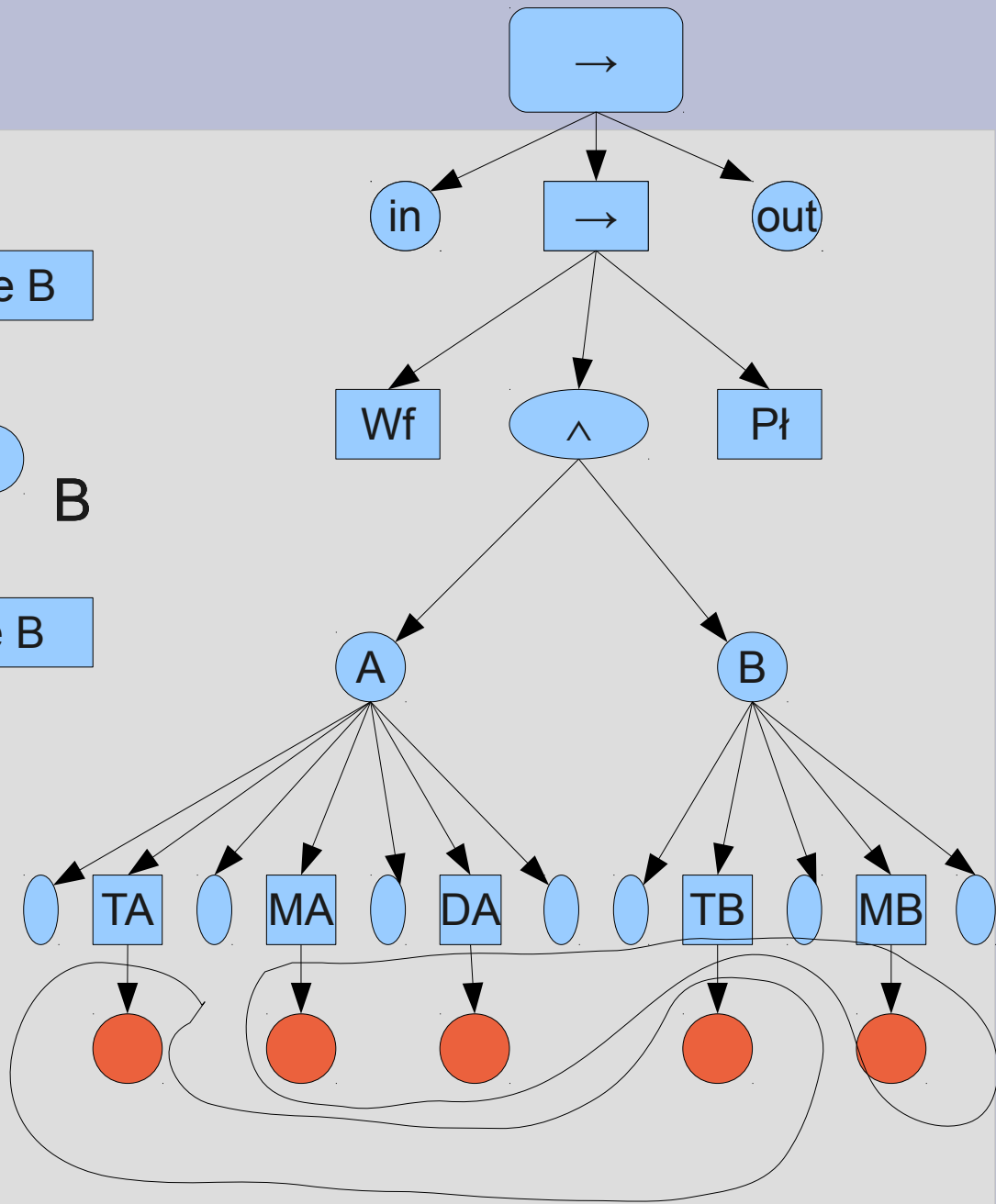
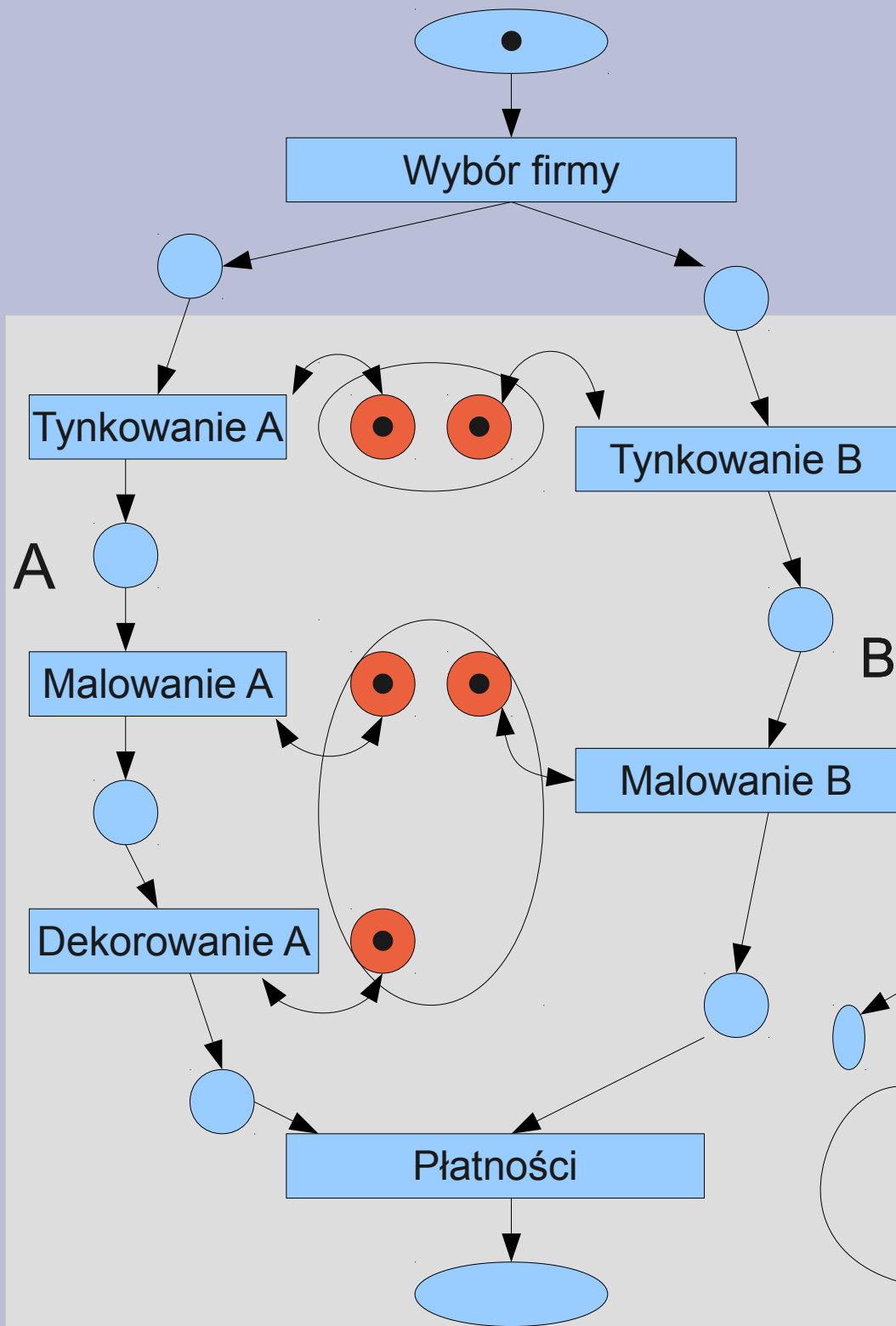


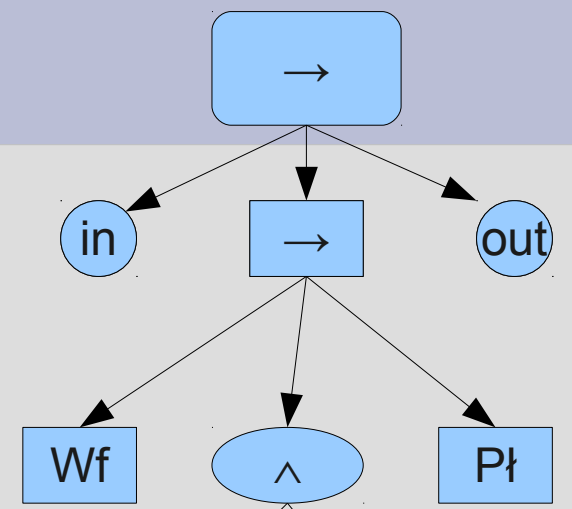
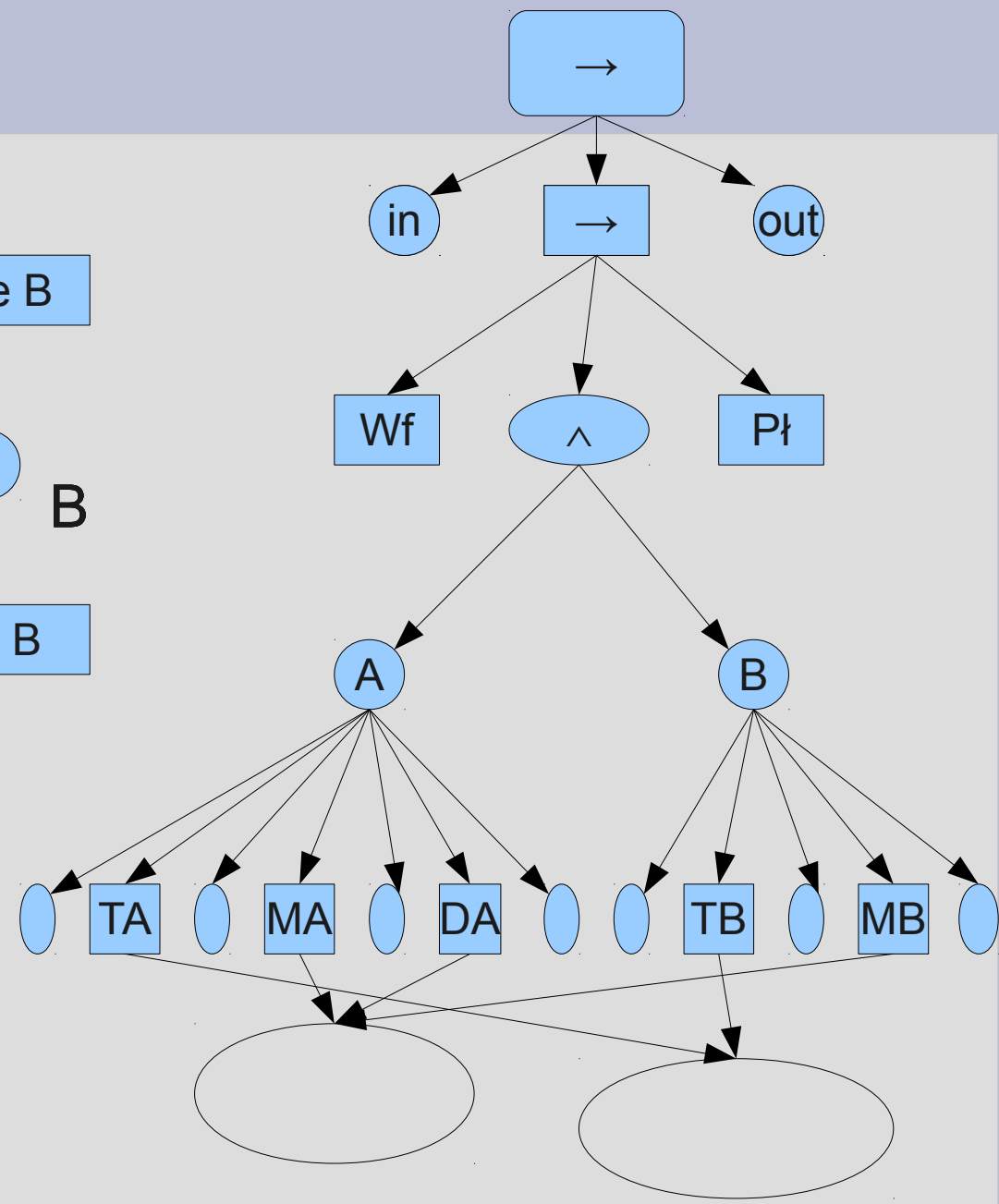
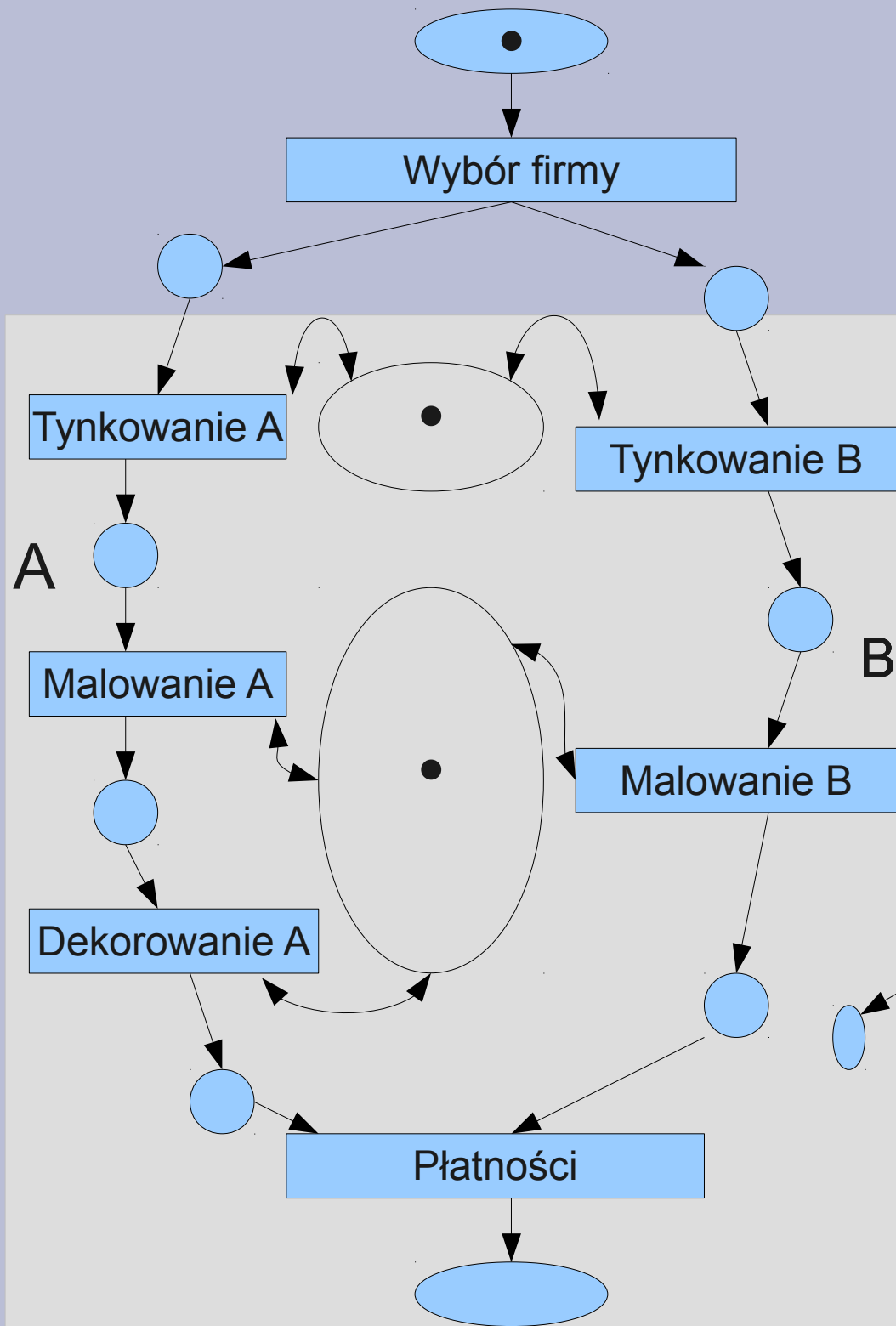


Drzewo rozdrobnień









Wyjaśnienie do poprzednich slajdów

- Pomarańczowe miejsca oznaczają zasoby: tynkarza, murarza i dekoratora.
- Dwaj pracownicy, którymi dysponuje firma, to tynkarz i malarz, z tym, że malarz ma umiejętności dekoratora. Stąd zespolenie w jedno kilku miejsc reprezentujących jeden zasób.
- Reguły pozwalające bezpiecznie (bez niebezpieczeństwa spowodowania blokady) łączyć miejsca zasobowe są nieco bardziej skomplikowane i nie będziemy się nimi tu zajmowali.

Drzewo rozdrobnień

- Bardzo użyteczne przy analizie.
- Po scaleniu sąsiadujących ze sobą i tak samo etykietowanych węzłów wewnętrznych jednoznacznie wyznaczone.
- Możliwe do szybkiego skonstruowania na podstawie grafu sieci (o ile w ogóle sieć da się otrzymać za pomocą rozdrobnień)
- Przechowujące bardzo ważną informację o strukturze procesu.
- Jednoznacznie wyznaczające sieć.

Zalety drzewa rozdrobnień

- W drzewie zapisana jest cała struktura logiczna procesu. Mamy historię jego tworzenia i informację o wzajemnych powiązaniach poszczególnych elementów.
- Drzewo umożliwia niezwykle prostą (również ze względów implementacyjnych) kategoryzację poszczególnych działań.

Węzły drzewa rozdrobnień

- Węzły drzewa odpowiadają odpowiednim poziomom abstrahowania.
 - umożliwiają powiązanie ze szczeblami zarządzania,
 - pozwalają w naturalny sposób określić prawa dostępu,
 - pozwalają wesprzeć organizację poufności danych.
 - Pozwalają w naturalny sposób określić obszary rażenia w sytuacjach awaryjnych

Wizualizacja

- Odpowiedni stopień zarządzania może mieć dostęp do informacji z dokładnością do pewnego poziomu rozdrobnień.
- Dyrektor generalny zapewne będzie chciał widzieć zgrubny obraz stanu procesu, czyli to, co znajduje się w górnych warstwach drzewa (bliżej korzenia :).
- Pracownik niższego szczebla nie musi znać szczegółów działek, które go nie dotyczą - wystarczy, że rozwinie sobie swój węzeł.
- Pracownikom można nadawać i odbierać uprawnienia do rozwijania węzłów.

Sytuacje awaryjne

- Wielkim problemem przepływów pracy są wymykające się schematom sytuacje awaryjne, których twórcy modelu nie przewidzieli
- Są one nie do uniknięcia, zwłaszcza jeśli realizacja przepływu trwa długo (miesiące, lata...)
- Często w takich sytuacjach przechodzi się na „ręczne sterowanie” i niekiedy jest to nieuniknione - nic nie zastąpi zdroworozsądkowo myślącego menedżera.

Sytuacje awaryjne (2)

- Jednym z podstawowych zadań, które pojawiają się w przypadku wystąpienia sytuacji awaryjnej jest
 - określenie obszaru rażenia
 - zaprojektowanie czynności ratunkowych (rollbacks, compensations)
 - zaproponowanie awaryjnej procedury dalszego postępowania.
 - jeśli się da, powrót do normalnego toku przetwarzania

Sytuacje awaryjne (3)

- Nie znam narzędzia, które w skuteczny sposób wspomagałyby sytuacje awaryjne.
- W szczególności trzeba rozważyć pomysł przeprojektowania fragmentu przepływu pracy, aby przystosować go do nowej sytuacji
- W trakcie takiego przeprojektowania można wprowadzić ciężkie błędy do dobrze pomyślanego projektu.

Drzewo rozdrobnień w sytuacjach awaryjnych

- Za pomocą drzewa rozdrobnień (a tak naprawdę tego, co jest naturalną konsekwencją podejścia strukturalnego) można w łatwy sposób
 - określić obszar rażenia
 - przeprojektować fragment sieci
 - bezpiecznie wrócić na tory

Określenie obszaru rażenia

- Zaczynamy od liści drzewa związanych z sytuacją awaryjną.
- Znajdujemy w drzewie najniższy węzeł będący wspólnym przodkiem tych liści.
- Potencjalny obszar rażenia, to jest wszystko to, co wypączkowało z tego węzła.
- Wszystko, co jest poza tym obszarem nie ma prawa być zainfekowane problemem, jeśli porządnie wyspecyfikowaliśmy cały przepływ pracy.

Przeprojektowanie

- W poddrzewie znalezionej węzła możemy stosując te same techniki dodawać lub usuwać węzły, strzałki bez obawy o utratę poprawności przepływu.
- Nawet jeśli menedżer nie ma pojęcia o projektowaniu, nie jest w stanie zepsuć przepływu stosując bezpieczne reguły.

Bezpieczny powrót na tory

- Sytuacja awaryjna dotyczy stanu, w którym w obszarze rażenia jest co najmniej jeden pionek (a zazwyczaj kilka).
- Reakcja polega najczęściej na wstrzymaniu wszystkich akcji i po poprawieniu schematu ponownym zamarkowaniu zainfekowanego obszaru - czasem w arbitralny sposób.
- Jak jednak to zrobić, żeby nowe - ad hoc wprowadzone markowanie - nie spowodowało np. zablokowania się sieci?

Poprawność zdarzenia ratunkowego

- Wprowadzamy nowy rodzaj zdarzeń - zdarzenia ratunkowe.
- Polegają one na zresetowaniu obszaru zainfekowanego awarią i ponownym jego zamarkowaniu w arbitralny sposób.
- Zdarzenia ratunkowe wiążemy z węzłami drzewa rozdrobnień - ściągają one pionki z całego zainfekowanego obszaru i redystrybuują je w wybranych miejscach.

Określanie poprawności zdarzeń ratunkowych

- Zdarzenie ratunkowe jest poprawne, gdy redystrybucja pionków jest w stanie doprowadzić nas do poprawnego markowania końcowego
- W sieciach budowanych ad hoc problem czy arbitralnie ustalone markowanie jest poprawne jest bardzo kosztowny (ogólny algorytm ma złożoność co najmniej podwójnie wykładniczą)
- W sieciach strukturalnych istnieje prosta metoda (czas działania liniowy) stwierdzania, czy markowanie jest poprawne.

Sieci strukturalne, a czas

- Za pomocą sieci strukturalnych można przeprowadzić w miarę prostą analizę czasową procesów stochastycznych związanych z wykonywaniem przepływów i przewidywać czasy zakończenia.
- Zazwyczaj stosuje się tu wariant stochastycznych sieci Petriego, w którym z każdym zdarzeniem związany jest czas wykonania danej akcji.

Sieci Petriego z czasem

- Zdarzeniom w sieciach Petriego (lub zasobom) przypisuje się najczęściej rozkład prawdopodobieństwa zmiennej opisującej czas wykonania.
- Wcale nieprostym problemem jest określenie rozkładu całości (lub części) przepływu pracy, gdy znamy wartości składowych.

Sieci stochastyczne

- Sieci stochastyczne, to sieci w których czasy wykonania opisują zmienne o rozkładzie wykładniczym - często odzwierciedlającym sytuacje w praktyce.
- Dla strukturalnych sieci stochastycznych zostały wyprowadzone wzory umożliwiające określenie rozkładu czasu wykonania (a więc i średniej i odchylenia standardowego i wszystkich innych potrzebnych parametrów) całości przepływu lub dowolnej jego części związanej z węzłem drzewa rozdobień.

Problemy ze strukturalnym podejściem

- Podejście strukturalne nie jest wystarczająco mocne, żeby modelować wszystkie sytuacje.
- Występujące w praktyce pewne ważne schematy są z natury rzeczy niestrukuralne.
- Ciekawą analizę przeprowadzili Bartosz Kiepuszewski, Wil van der Aalst i Artur ter Hofstede, którzy na początku tego wieku sklasyfikowali podstawowe wzorce cegiełek, z których buduje się przepływy pracy.

Workflow patterns

- Wyniki prac wspomnianego zespołu (w tym praca doktorska Kiepuszewskiego) stanowią doskonały punkt odniesienia, dając jednocześnie wskazówki, czego w rozważanym modelu (narzędziu) brakuje.
- Nasz model daje się uzupełnić o pewne dodatkowe, niestrukturalne narzędzia (jak komunikacja między równoległymi wątkami), które ratują większość sytuacji, nadal zachowując poprawność.

Co się da zrobić metodą strukturalną?

- Zaproponowane podejście z zasygnalizowanymi rozszerzeniami ma podobną siłę wyrazu, jak dość popularny język specyfikacji przepływów pracy BPEL.
- Zastrzeżenie: w BPELu da się projektować niepoprawne schematy, tutaj nie!

Poważniejszy problem

- Brak narzędzi
- Istnieje parę prototypów, które są realizowane na UW, ale bardziej jako badawcza weryfikacja teoretycznych pomysłów, niż jako narzędzie nadające się do wdrożenia

Jeszcze poważniejszy problem

- Podejście strukturalne wymaga zmiany myślenia
- Trzeba przyzwyczaić się do myślenia metodą zstępującą (top-down), a nie czymś, co przypomina przechodzenie grafu wgłąb (depth-first-search)!
- To się powinno opłacić!
- Programistom się udało!

Podsumowanie

- Wady i zalety podejścia strukturalnego.
- Wady:
 - wymagana zmiana sposobu myślenia
 - nie do końca jasne, czy wszystkie ważne wzorce da się wyrazić
 - brak narzędzi
- Zalety:
 - strukturalizacja projektu
 - związanie ról zarządczych ze strukturą
 - ułatwienie dynamicznej obsługi awarii
 - zapewnienie poprawności schematów
 - ułatwienia implementacyjne (czasem istotne!)
 - elastyczność i skalowalność
 - gospodarka zasobami zintegrowana

Przesłanie

Dawniej też się ludziom wydawało, że program może liczyć najwyżej parę tysięcy instrukcji!

© Piotr Chrzastowski-Wachtel