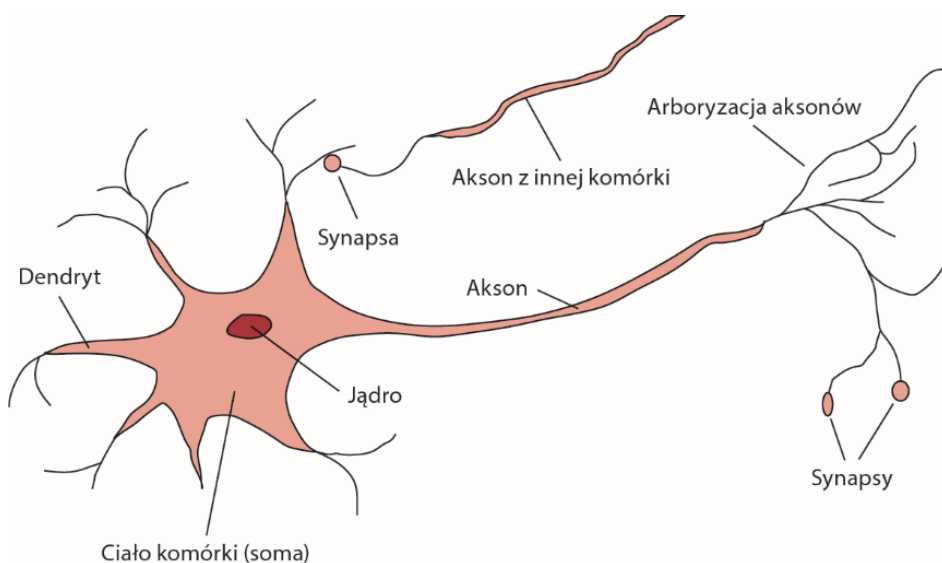
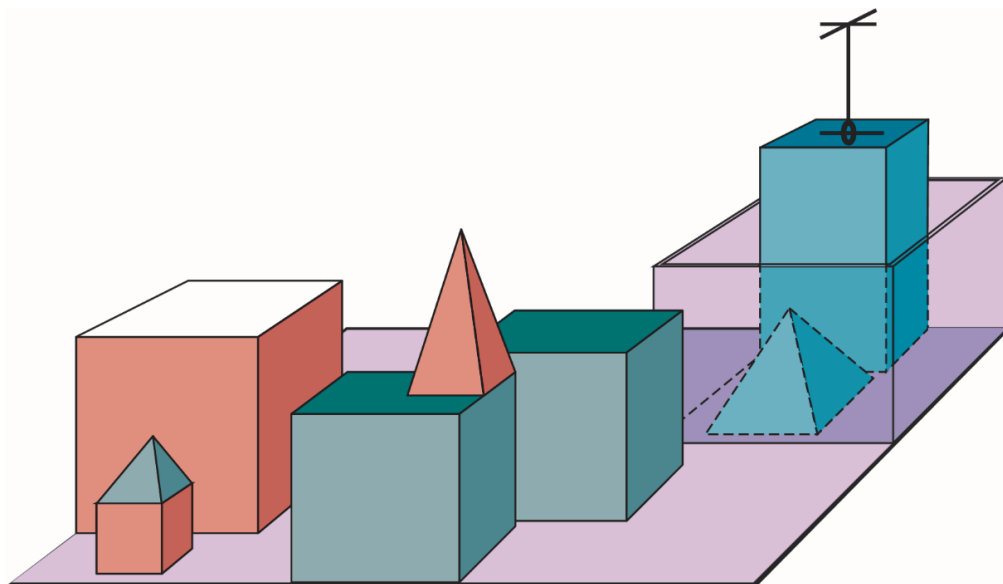


## ROZDZIAŁ 1

# WSTĘP



**RYSUNEK 1.1.** Części komórki nerwowej (neuronu). Każdy neuron składa się z ciała komórkowego (somy), które zawiera jądro komórkowe. Z ciała komórki rozgałęziają się włókna zwane dendrytami i jedno długie włókno zwane aksonem. Akson rozciąga się na dużą odległość, znacznie większą niż wskazuje skala na tym rysunku. Akson ma zwykle 1 cm długości (to 100-krotność średnicy ciała komórki), ale może osiągnąć nawet 1 metr. Neuron tworzy połączenia z innymi neuronami (w liczbie 10 – 100 000) za pomocą łączników zwanych synapsami. Sygnały są przenoszone między neuronami w wyniku skomplikowanych reakcji elektrochemicznych. Sygnały kontrolują aktywność mózgu w krótkim okresie, umożliwiają także długoterminowe zmiany w połączeniach neuronów. Uważa się, że mechanizmy te stanowią podstawę procesu uczenia się mózgu. Większość przetwarzania informacji zachodzi w korze mózgowej — zewnętrznej warstwie mózgu. Podstawową jednostką organizacyjną wydaje się kolumna tkanki o średnicy około 0,5 mm, zawierająca około 20 000 neuronów i rozciągająca się na całą głębokość kory (u ludzi około 4 mm)

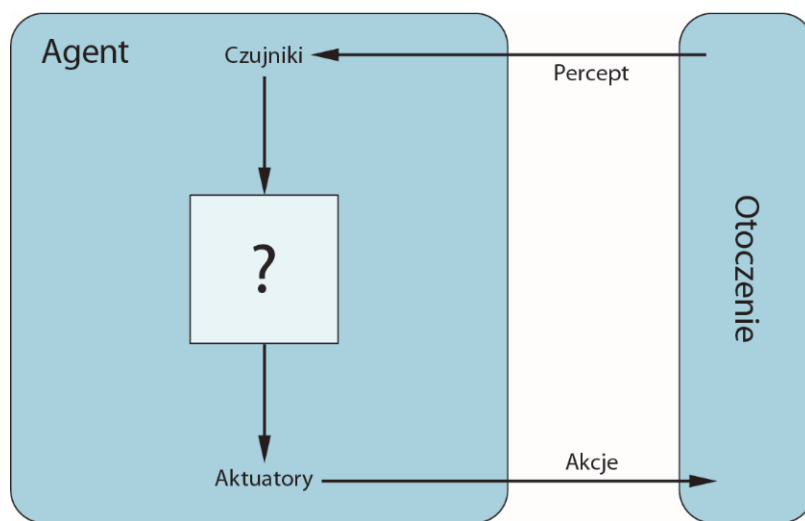


**RYSUNEK 1.2.** Scena ze świata blokowego: program SHRDLU (Winograd, 1972) właśnie zakończył wykonywanie polecenia „Znajdź blok wyższy od tego, który aktualnie trzymasz, i włóż go do pudełka”

---

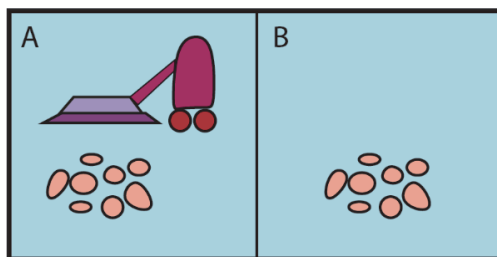
## ROZDZIAŁ 2

# INTELIGENTNI AGENCI<sup>1</sup>



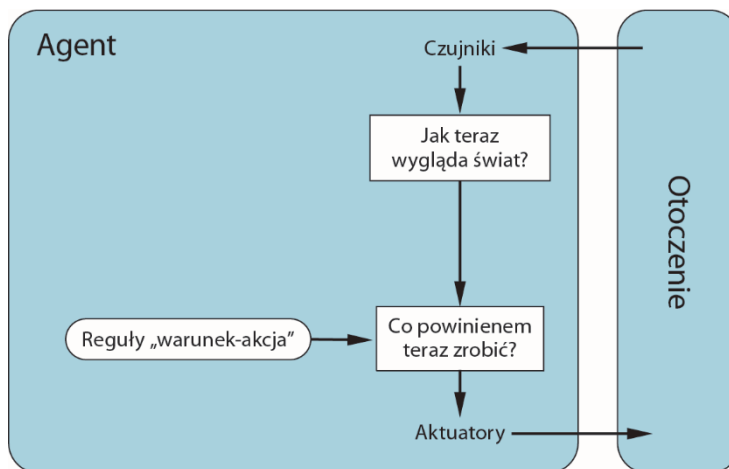
RYSUNEK 2.1. Interakcja agenta ze środowiskiem poprzez czujniki i aktuatory

<sup>1</sup> Ponieważ agentem może być zarówno maszyna, jak i człowiek, będę w tej książce używać formy osobowej — „agenci”, nie „agenty” — *przyp. tłum.*



**RYSUNEK 2.2.** Świat odkurzacza z dwiema lokalizacjami, z których każda może być czysta albo brudna. Odkurzacza może przemieszczać się w lewo lub prawo i odkurzać lokalizację, w której obecnie się znajduje. W różnych konfiguracjach świata odkurzaczy obowiązywać mogą różne reguły percepcji, samooceny robota pod względem powodzenia zamierzonej akcji itp.

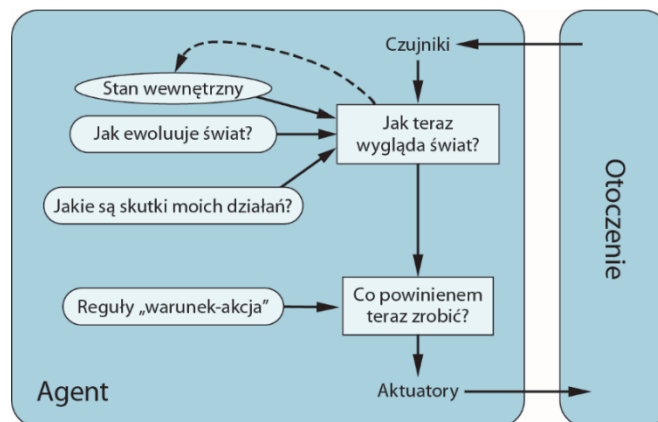
---



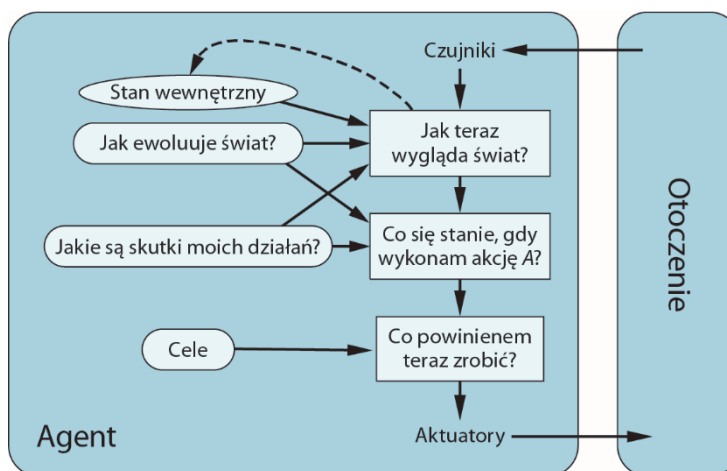
**RYSUNEK 2.3.** Schematyczny diagram funkcjonowania prostego agenta odruchowego. Prostokąty reprezentują elementy wewnętrznego stanu procesu decyzyjnego agenta, owalne ramki przedstawiają informację wykorzystywaną w tym procesie

---

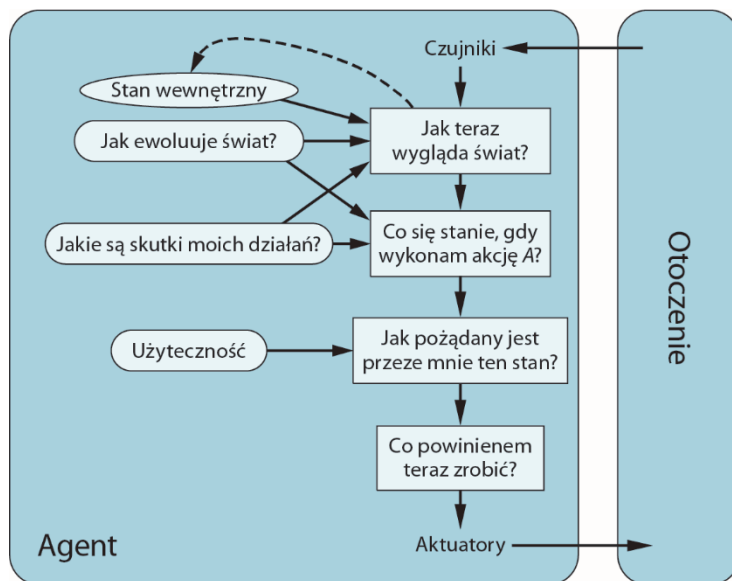




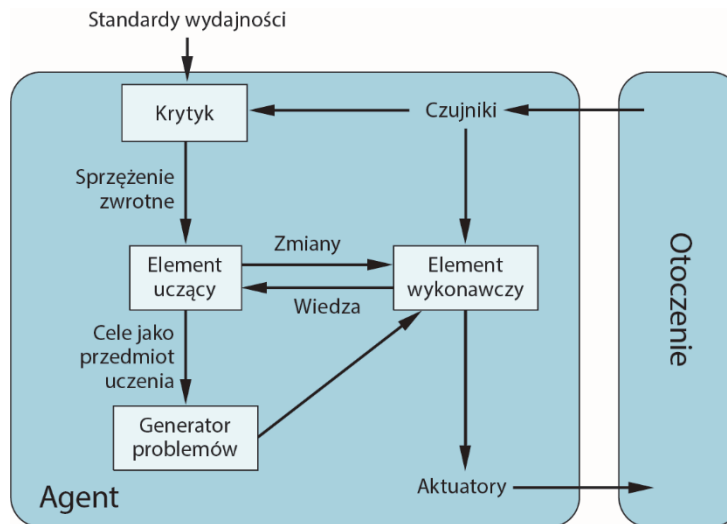
RYSUNEK 2.4. Schemat agenta odruchowego opartego na modelach



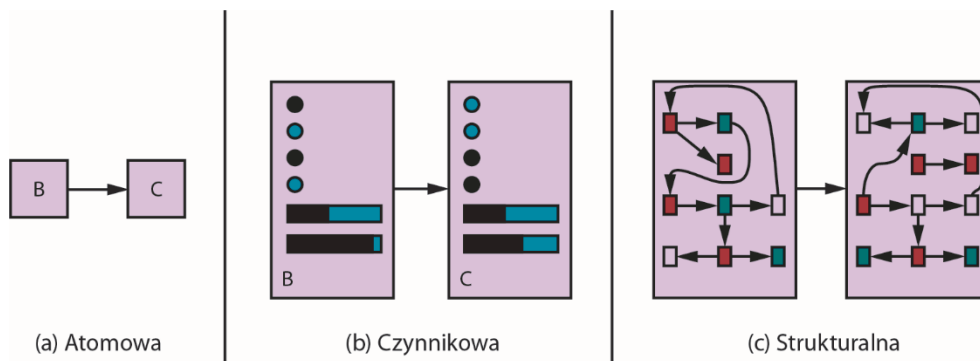
RYSUNEK 2.5. Schemat działania agenta ukierunkowanego na cel. Agent ten bierze pod uwagę zarówno informacje o stanie otaczającego świata, jak i zbiór celów, które zamierza osiągnąć, i podejmuje akcje, które powinny doprowadzić do urzeczywistnienia tego zamierzenia



**RYSUNEK 2.6.** Schemat agenta opartego na modelach, kierującego się względami użyteczności. Taki agent wykorzystuje model otaczającego świata w połączeniu z funkcją użyteczności, która ilościowo ujmuje jego własne preferencje odnośnie stanów tego świata. Następnie decyduje się on na akcję, która — zgodnie z tą funkcją — maksymalizuje oczekiwaną użyteczność, rozumianą jako ważone uśrednienie po wszystkich możliwych wynikach, przy czym wagami są prawdopodobieństwa zaistnienia poszczególnych wyników



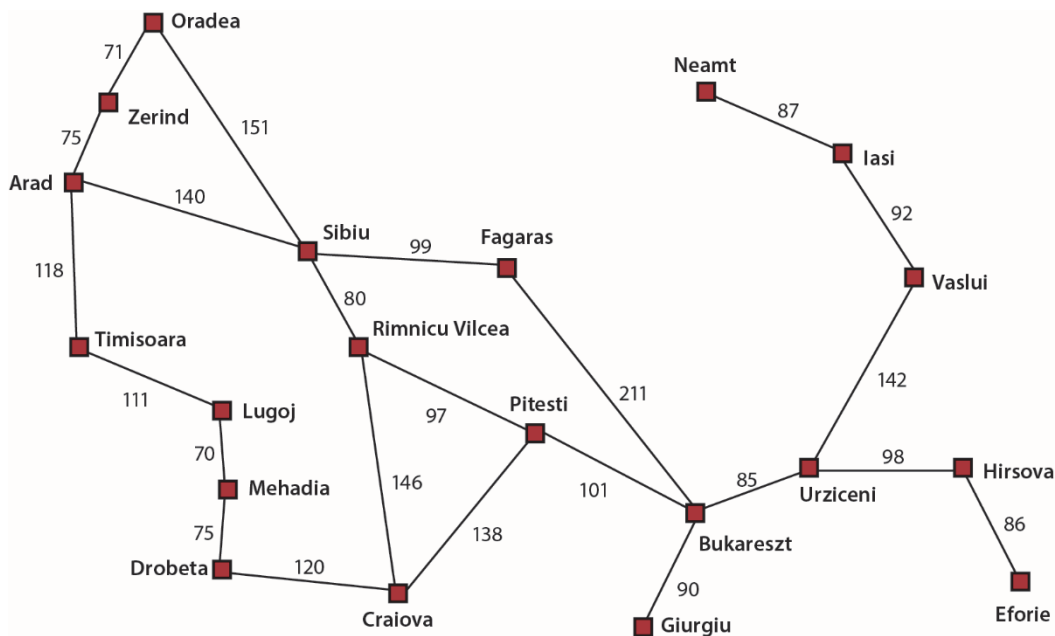
**RYSUNEK 2.7.** Ogólny schemat agenta uczącego się. Element wykonawczy reprezentuje to, co dotychczas utożsamialiśmy z programem agenta. Zadaniem elementu uczącego jest modyfikowanie tego programu w celu osiągnięcia lepszej wydajności



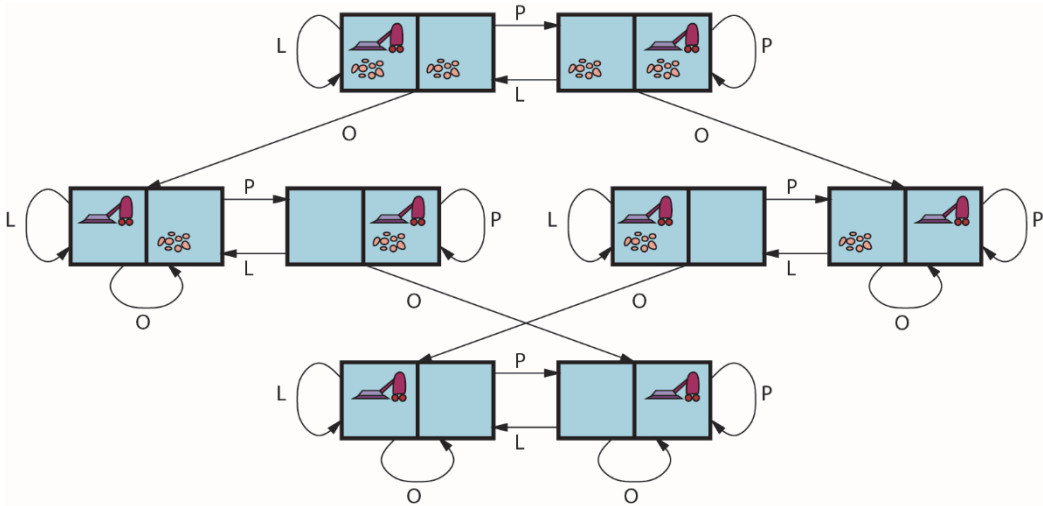
**RYSUNEK 2.8.** Trzy sposoby reprezentowania stanu środowiska i przejść między nimi. W reprezentacji atomowej (a) stan (taki jak *B* lub *C*) jest czarną skrzynką bez wewnętrznej struktury. Czynnikiowa reprezentacja stanu (b) jest wektorem wartości atrybutów — wartości te mogą być liczbami rzeczywistymi, wartościami boolowskimi lub ustalonymi zbiorami symboli. W reprezentacji strukturalnej (c) stan ma postać zbioru obiektów, z których każdy posiada własne atrybuty i odniesienia do innych obiektów

## ROZDZIAŁ 3

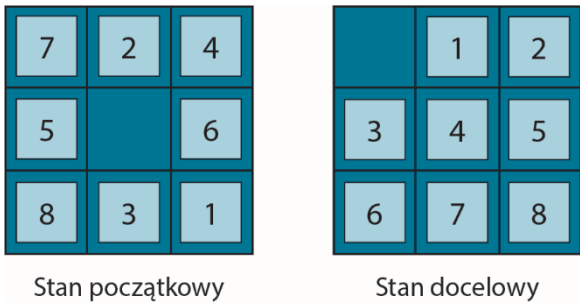
# ROZWIĄZYWANIE PROBLEMÓW ZA POMOCĄ WYSZUKIWANIA



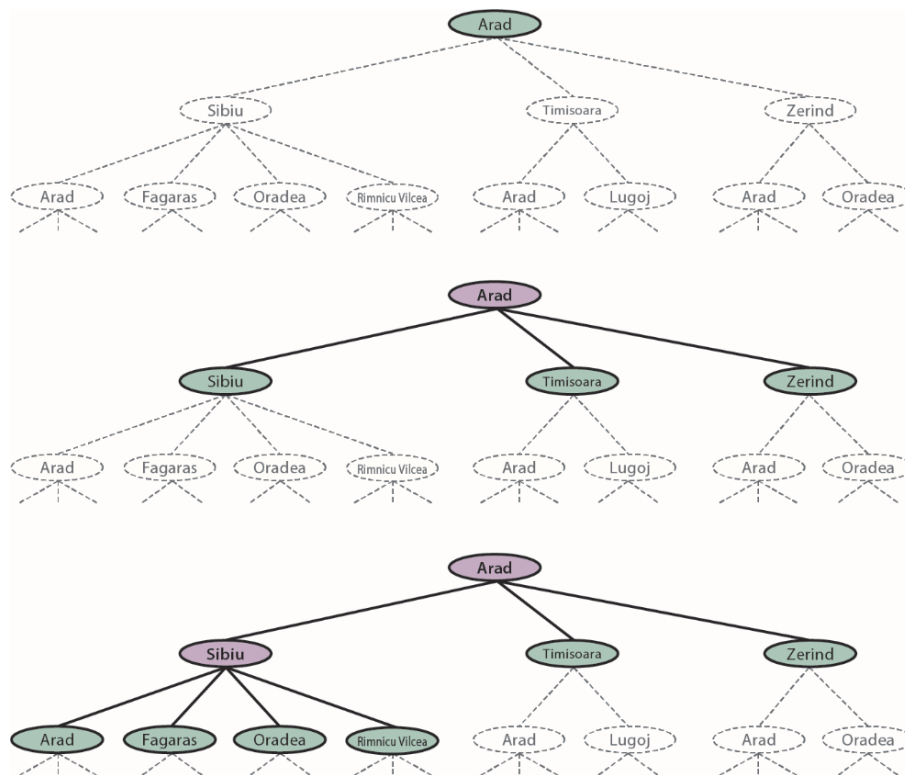
RYSUNEK 3.1. Uproszczona mapa części Rumunii, z zaznaczonymi odległościami (w milach) między miastami



RYSUNEK 3.2. Graf przestrzeni stanów dla dwukomórkowego środowiska odkurzacza, uwidoczniający 8 stanów i trzy akcje: L — *W\_lewo*, P — *W\_prawo*, O — *Odkurz*



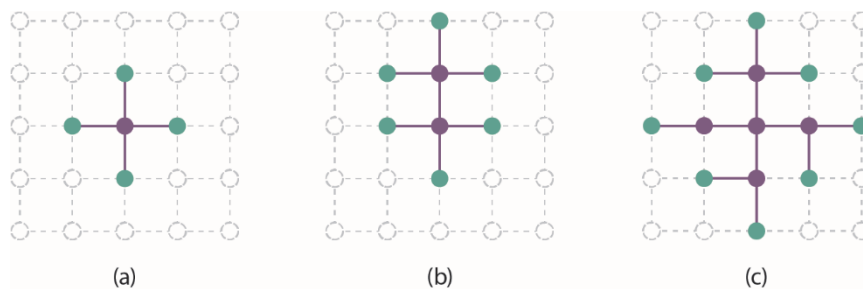
RYSUNEK 3.3. Przykładowa instancja gry puzzle-8



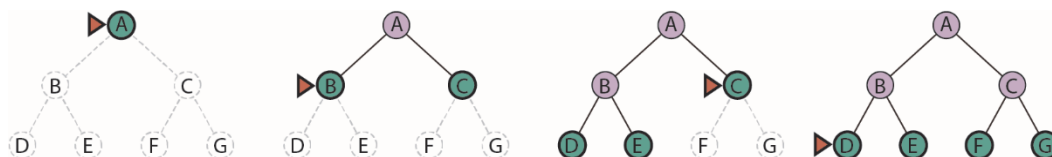
**RYСУNEK 3.4.** Trzy częściowe drzewa wyszukiwawcze odpowiadające poszukiwaniu drogi z Arad do Bukaresztu. Węzły rozwinięte mają kolor fioletowy i pogrubioną czcionkę w etykiecie, węzły graniczne, wygenerowane, ale jeszcze nie rozwinięte, mają kolor zielony i etykietowane są przy użyciu zwykłej czcionki. Stany odpowiadające obu tym rodzajom węzłów nazywane są stanami osiągniętymi. Węzły, które mogą zostać wygenerowane jako kolejne, otoczone są cienką linią przerywaną. Zauważmy, że w dolnym drzewie występuje cykliczne następstwo stanów Arad – Sibiu – Arad, wykluczające optymalność ścieżki, zatem ścieżka ta nie powinna być dalej rozwijana



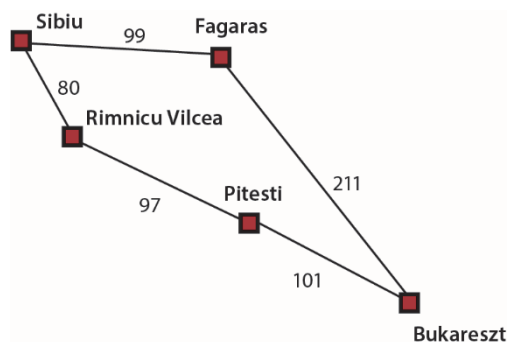
**RYСУNEK 3.5.** Sekwencja wybranych etapów rozwojowych drzewa wyszukiwawczego generowanego przez graf z rysunku 3.1. Na każdym etapie rozwijamy wszystkie węzły tworzące granicę, rozszerzając w ten sposób ścieżki, ograniczając się do akcji, które mają sens i nie prowadzą do któregoś ze stanów już osiągniętych. Zauważmy, że w trzecim etapie rozwinięcie węzła Oradea doprowadziłoby do wygenerowania dwóch węzłów potomnych, reprezentujących stany już osiągnięte (Zerind i Sibiu), dlatego rezygnujemy z tego rozwinięcia



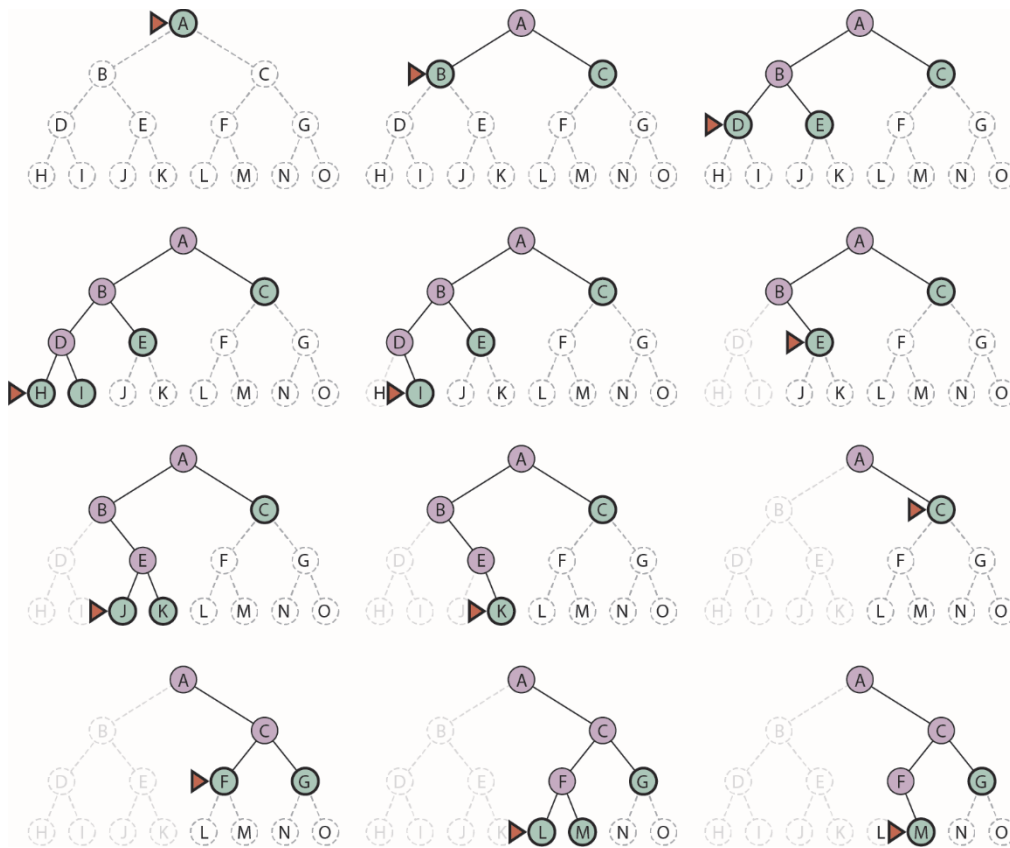
**RYSUNEK 3.6.** Zjawisko separacji przy przeszukiwaniu grafu, zilustrowane za pomocą prostokątnej siatki (gridu). Granica (kolor zielony) oddziela region wewnętrzny (kolor fioletowy) od zewnętrznego (brak koloru). Granica stanowi zbiór węzłów reprezentujących stany osiągnięte, ale jeszcze nierozwiniętych. Region wewnętrzny to zbiór węzłów już rozwiniętych, węzły w regionie zewnętrznym reprezentują natomiast stany jeszcze nieosiągnięte. W stadium (a) rozwinięty jest jedynie korzeń drzewa, w stadium (b) rozwinięty jest najwyższy położony węzeł granicy, stadium (c) to efekt rozwinięcia pozostałych węzłów potomnych korzenia, odwiedzanych w kolejności zgodnej z ruchem wskazówek zegara



**RYSUNEK 3.7.** Wyszukiwanie wszerek w prostym drzewie binarnym. Na każdym etapie węzeł przeznaczony do rozwinięcia zaznaczony jest trójkątnym wskaźnikiem

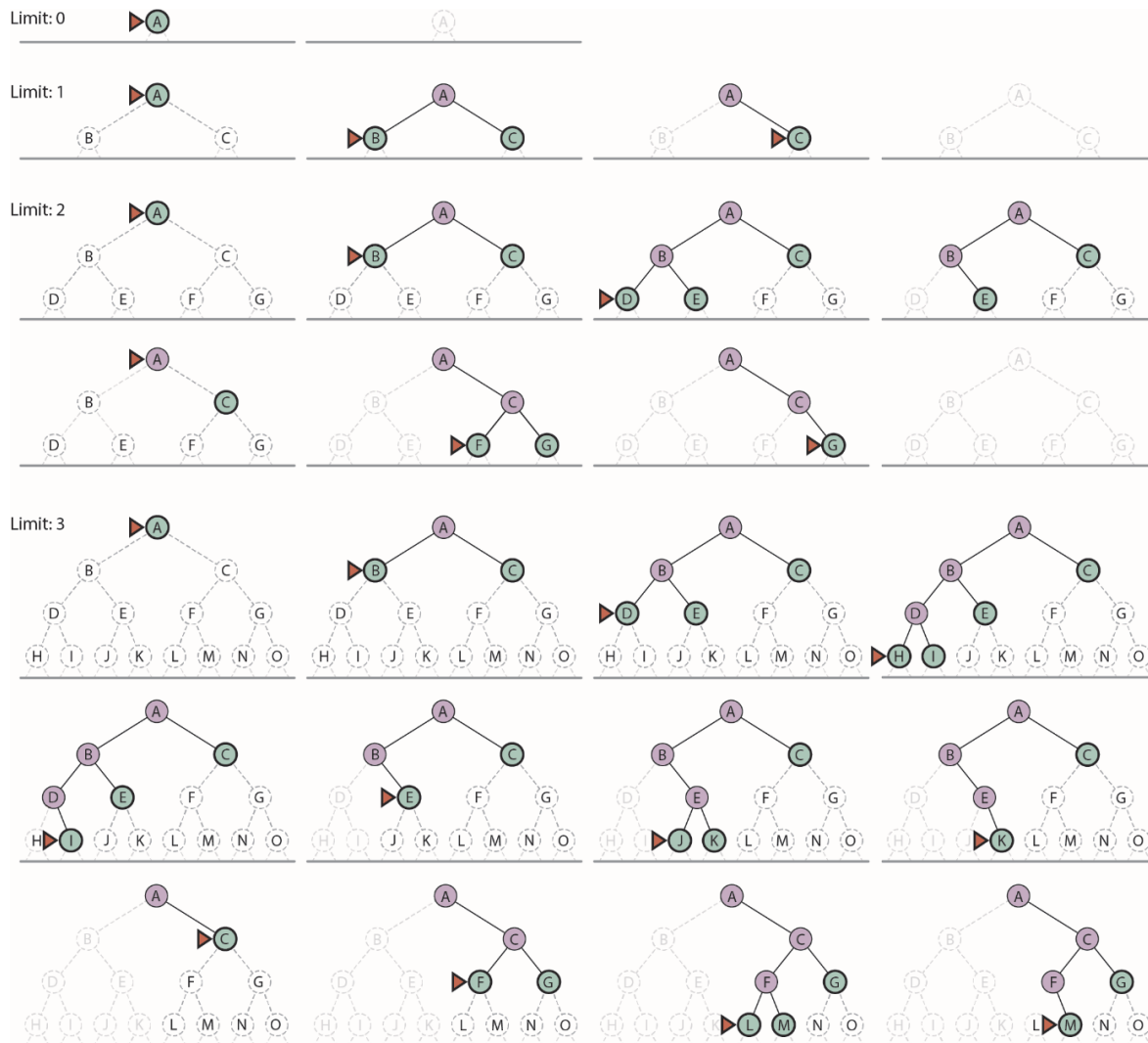


**RYSUNEK 3.8.** Część przestrzeni stanów podróży po Rumunii, wybrana dla zilustrowania wyszukiwania przy jednolitym koszcie

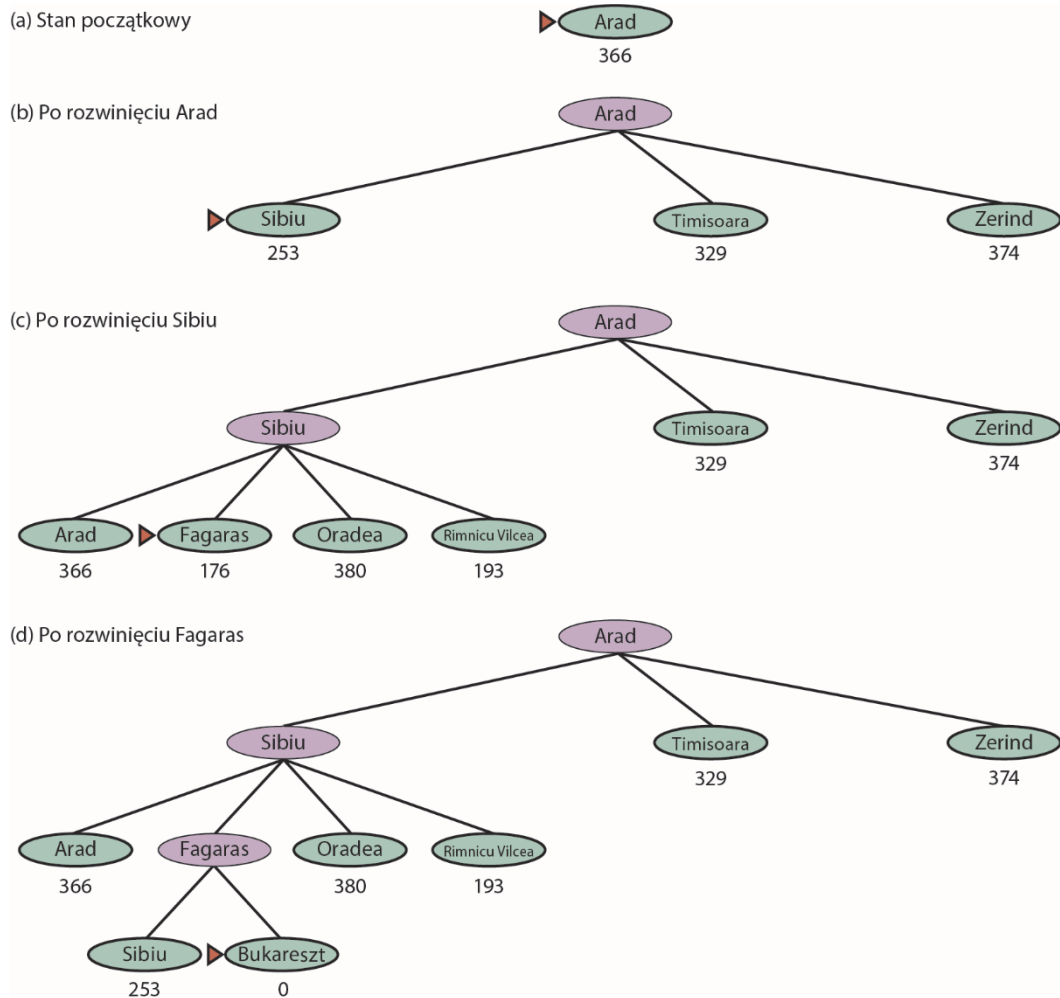


**RYSUNEK 3.9.** Dwanaście kroków (od lewej do prawej, z góry na dół) przebiegu wyszukiwania w głąb w drzewie binarnym, począwszy od stanu A, skończywszy na stanie docelowym M. Kolorem zielonym oznaczono węzły tworzące bieżącą granicę, trójkątny wskaźnik identyfikuje węzeł do aktualnego rozwijania. Węzły już rozwinięte oznaczone są kolorem fioletowym, węzły stanowiące kandydatury do następnych rozwinięć są niepokolorowane. Słabiej widoczne rozwinięte węzły niemające węzłów potomnych w granicy wykluczane są z dalszych poszukiwań

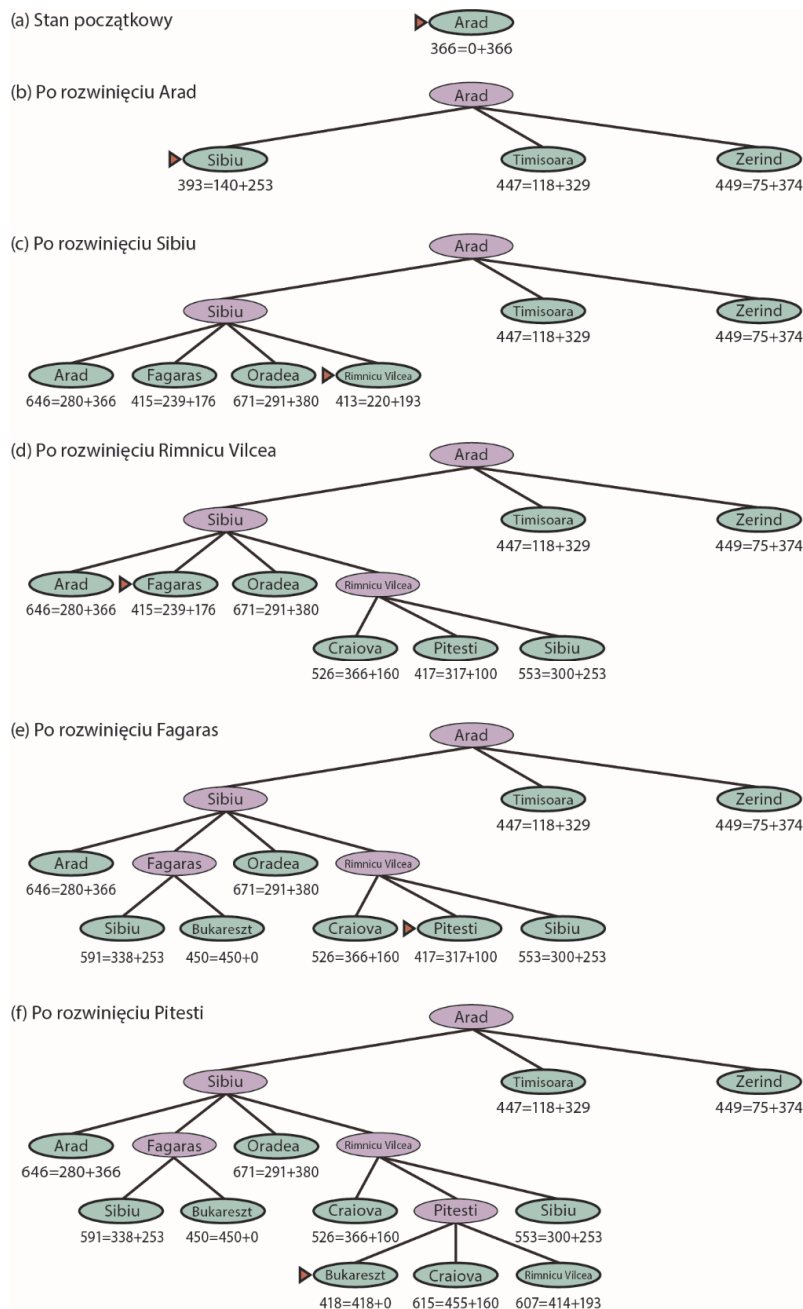




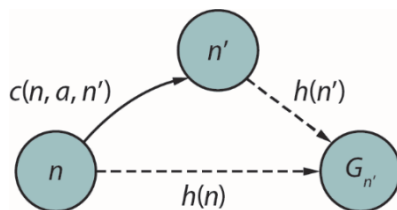
**RYSUNEK 3.10.** Cztery iteracje wyszukiwania z zagłębieniem w drzewie binarnym, z limitem głębokości zmieniającym się od 0 do 3, stan docelowy reprezentowany jest przez węzeł M. Zauważmy, że węzły wewnętrzne tworzą pojedynczą ścieżkę. Trójkąty wskazują węzły do rozwinięcia; zielone węzły z ciemną obwódką tworzą granicę, węzły słabiej widoczne wykluczone są jako rozwiązania na danym poziomie zagłębienia



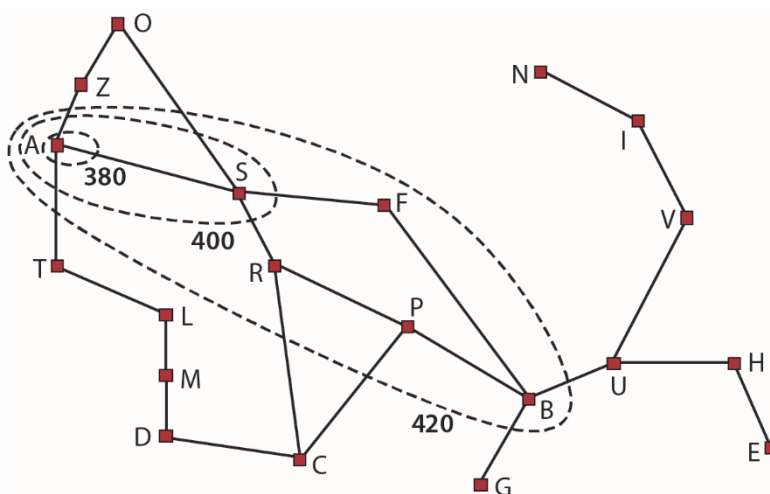
**RYSUNEK 3.11.** Kolejne etapy znajdowania drogi z Arad do Bukaresztu metodą zachłannego wyszukiwania drzewiastego „najpierw najlepszy” przy zastosowaniu heurystyki  $h_{SLD}$ . Dla poszczególnych węzłów podano wartości tej heurystyki



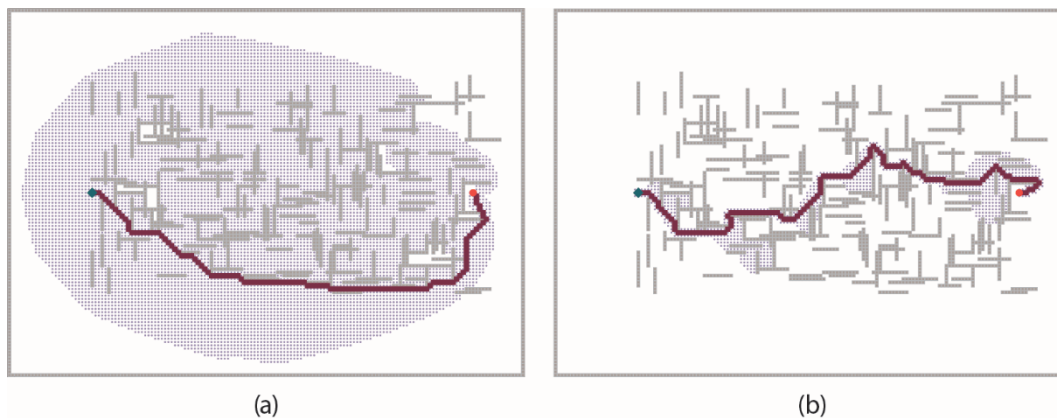
RYСУNEK 3.12. Kolejne etapy wyszukiwania A\* znajdującego drogę do Bukaresztu. Dla poszczególnych węzłów podano wartości  $f = g + h$ , gdzie  $h$  jest odległością w linii prostej od celu, odczytaną z tabeli 3.2



**RYSUNEK 3.13.** Nierówność trójkąta ilustrująca spójność heurystyki: dla węzła  $n$  i jego następnika  $n'$  generowanego przez akcję  $a$  wartość  $h(n)$  nie może być większa od sumy  $h(n')$  i kosztu akcji  $a$

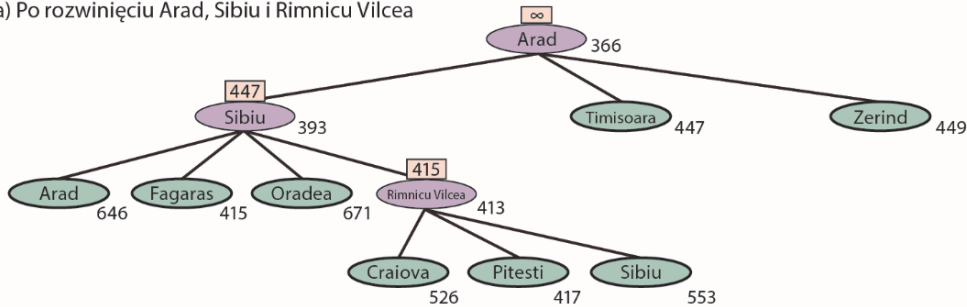


**RYSUNEK 3.14.** Mapa Rumunii z rysunku 3.1 z uwidocznionymi konturami dla  $f = 380$ ,  $f = 400$  i  $f = 420$ , z węzłem początkowym Arad. Dla węzłów znajdujących się wewnątrz konturu etykietowanego wartością  $w$  zachodzi zależność:  $f + g \leq w$

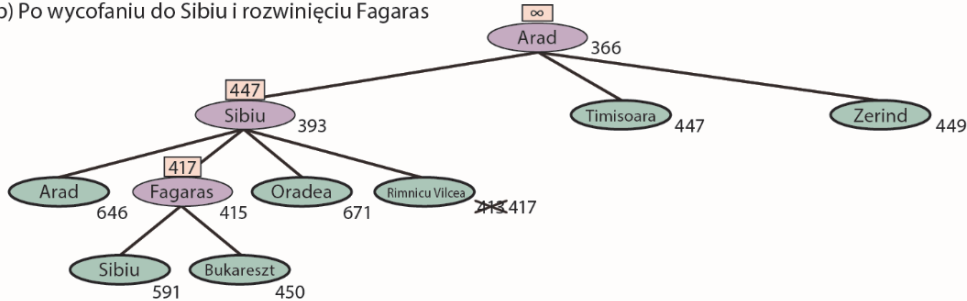


**RYSUNEK 3.15.** Rezultaty dwóch wyszukiwań na tym samym gridzie: (a) klasyczne wyszukiwanie A\* (b) ważne wyszukiwanie A\* z wagą  $W = 2$ . Szare paski to przeszkody między komórkami, purpurowa linia to ścieżka od stanu początkowego (zielony punkt) do stanu docelowego (czerwony punkt). Małe kropki reprezentują stany osiągnięte w procesie wyszukiwania. W tym konkretnym przypadku ważne wyszukiwanie A\* przeanalizowało 7-krotnie mniej stanów i zwróciło rozwiązanie droższe o 5% w porównaniu z klasycznym wyszukiwaniem A\*.

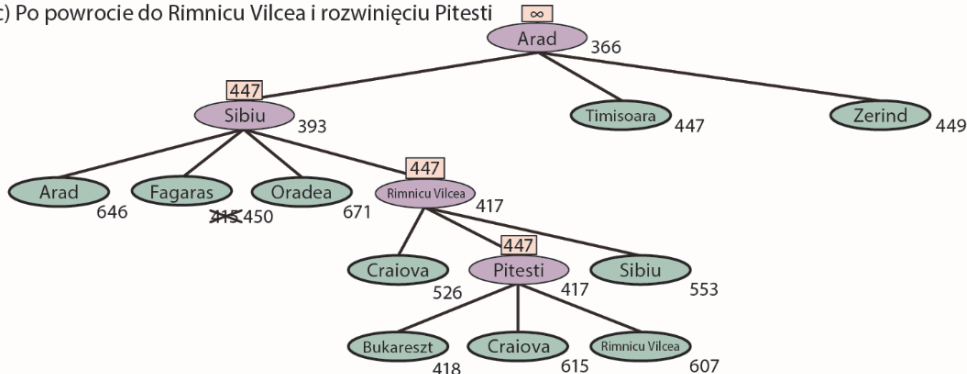
(a) Po rozwinięciu Arad, Sibiu i Rimnicu Vilcea



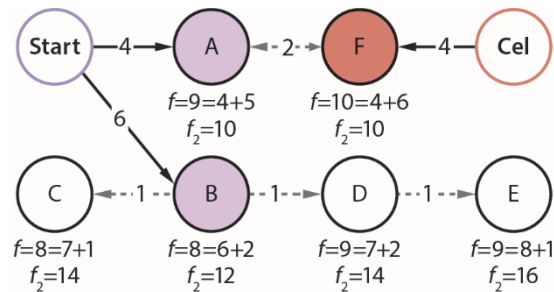
(b) Po wycofaniu do Sibiu i rozwinięciu Fagaras



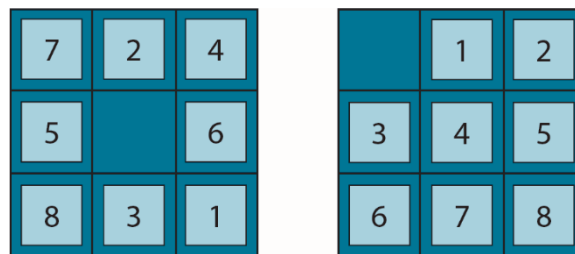
(c) Po powrocie do Rimnicu Vilcea i rozwinięciu Pitesti



**RYСУNEK 3.16.** Wyszukiwanie RBFS najkrótszej ścieżki do Bukaresztu. Nad bieżącymi węzłami widoczne są wartości zmiennej  $limit\_f$  zwracanej przez ostatnie wywołanie rekurencyjne, a każdy węzeł etykietowany jest wartością funkcji  $f$ . Etap (a): ścieżka przez Rimnicu Vilcea jest rozwijana aż do wystąpienia sytuacji, gdy najlepszy liść (Pitesti) ma wartość gorszą od najlepszej ścieżki alternatywnej (Fagaras). Etap (b): rekurencja cofa się i wartość  $f$  najlepszego liścia „zapomnianego” poddrzewa (417) jest archiwizowana w węźle Rimnicu Vilcea. Następnie rozwijany jest węzeł Fagaras, co prowadzi do wykrycia wartości 450 jako najlepszej wśród liści. Etap (c): rekurencja cofa się i wartość najlepszego liścia „zapomnianego” poddrzewa (450) jest archiwizowana w węźle Fagaras, po czym rozwijany jest węzeł Rimnicu Vilcea. Ponieważ w tym momencie koszt najlepszej ścieżki alternatywnej (przez Timisoara) jest równy co najmniej 447 (czyli więcej niż 417), rozwinięcie węzła Rimnicu Vilcea prowadzi do Bukaresztu



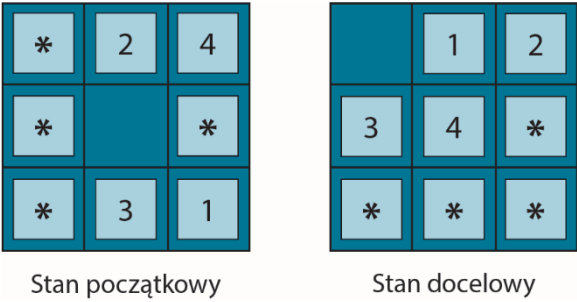
**RYСУNEK 3.17.** Dwukierunkowe wyszukiwanie utrzymuje dwie kolejki węzłów. Z lewej strony węzły A i B są następnikami węzła początkowego Start; z prawej strony węzeł F jest wstecznym następnikiem węzła Cel. Dla każdego węzła podano wartość funkcji ewaluacyjnej  $f = g + h$  i wartość  $f = \max(2g, g + h)$ . Wartością funkcji  $g$  jest suma kosztów akcji, uwidocznionych przy poszczególnych strzałkach, wartości heurystyki  $h$  przyjęto dowolnie (nie da się ich wydedukować z rysunku). Koszt optymalnego rozwiązania Start-A-F-Cel wynosi  $C^* = 4 + 2 + 4 = 10$ , co oznacza, że dwukierunkowy algorytm „spotkania pośrodku” nie powinien rozwijać żadnego węzła, dla którego  $g > \frac{C^*}{2} = 5$ , więc następnym rozwijanym węzłem na drodze do optymalnego rozwiązania powinien być A albo F (dla obu  $g = 4$ ). Gdybyśmy kierowali się najmniejszą wartością funkcji  $f$ , wybralibyśmy do rozwinięcia węzły B i C ( $f = 8$ ) zamiast A ( $f = 9$ ), następnie D i E zostałyby związane z A. Dla nich wszystkich jednak  $g > \frac{C^*}{2}$ , więc żaden z nich nie zostanie rozwinięty, gdy funkcją ewaluacyjną będzie  $f_2$ .



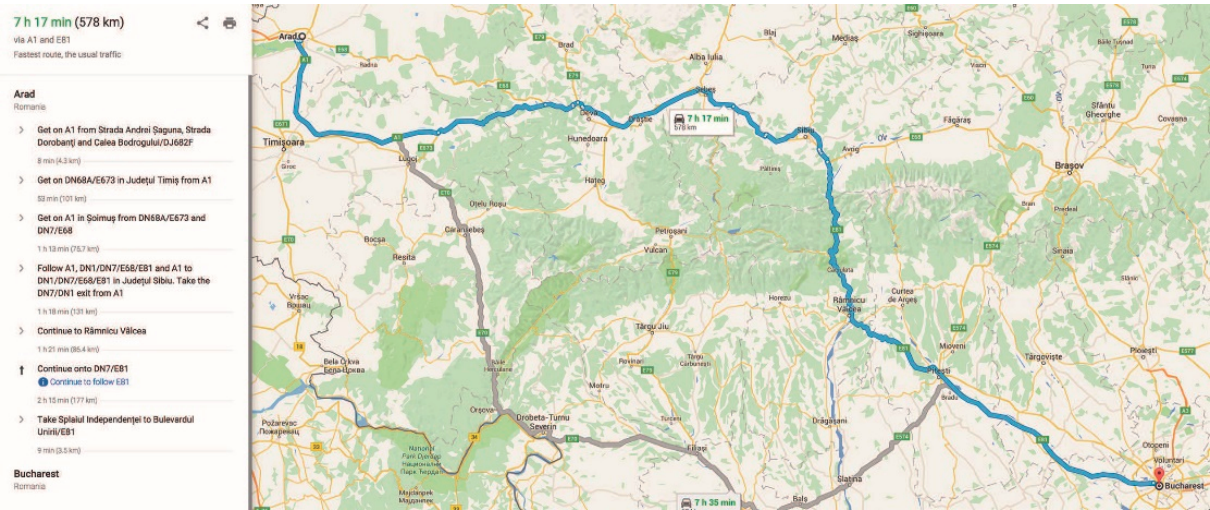
Stan początkowy

Stan docelowy

**RYСУNEK 3.18.** Jedna z typowych instancji gry puzzle-8, najkrótsze rozwiązanie wymaga 26 akcji



RYSUNEK 3.19. Instancja podproblemu układanki puzzle-8 z rysunku 3.18 polegająca na sprowadzeniu na docelowe lokalizacje kafelków 1, 2, 3, 4i pustego miejsca, bez zajmowania się innymi kafelkami

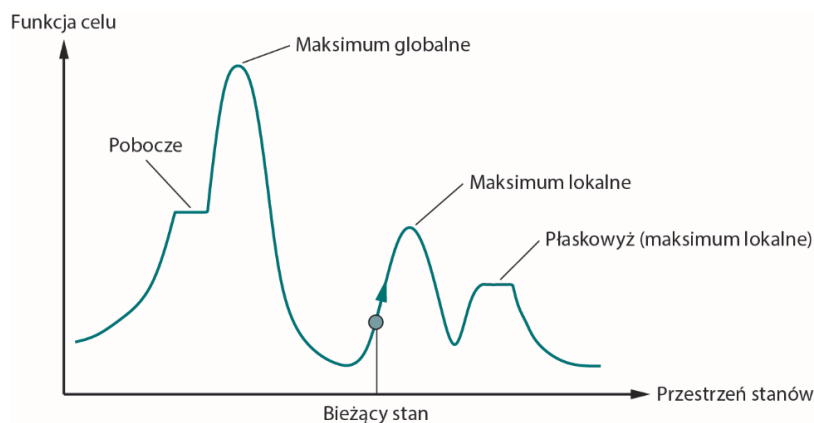


RYSUNEK 3.20. Usługa webowa umożliwiająca wytyczanie tras, funkcjonująca w oparciu o algorytmy wyszukiwania

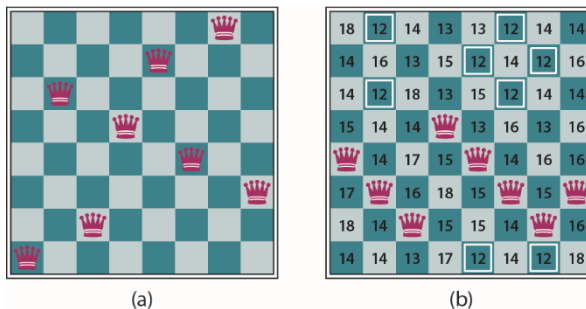


## ROZDZIAŁ 4

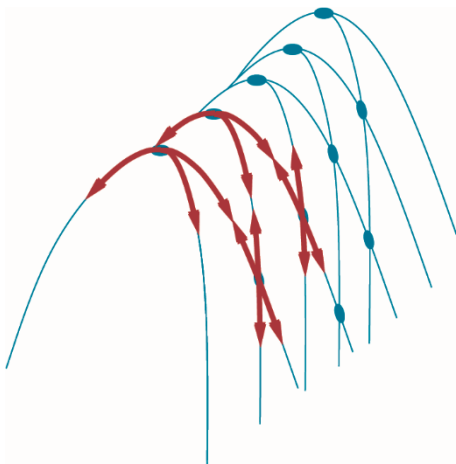
# WYSZUKIWANIE W ZŁOŻONYCH ŚRODOWISKACH



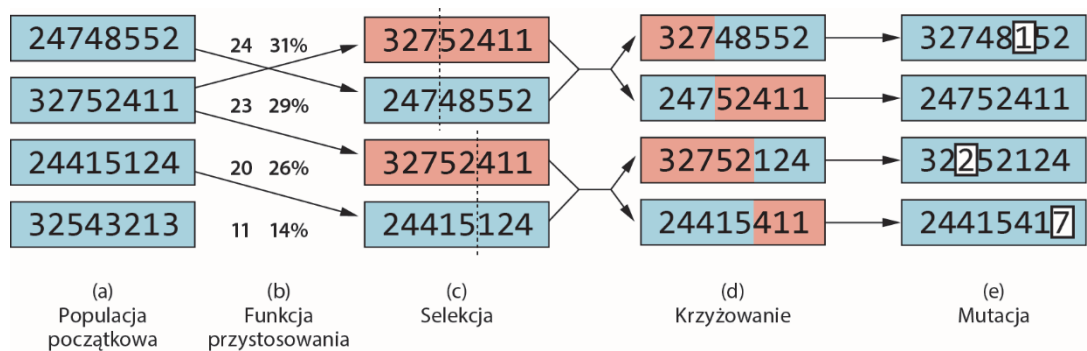
**RYSUNEK 4.1.** Krajobraz jednowymiarowej przestrzeni stanów, w której wartość funkcji celu odpowiada wzniesieniu. Celem problemu jest znalezienie globalnego maksimum



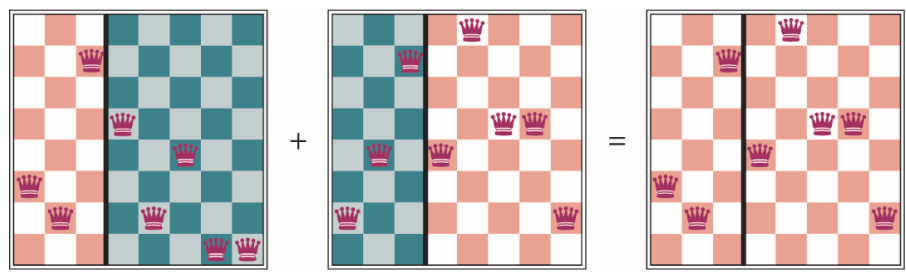
**RYСУNEK 4.2.** Problem ośmiu hetmanów — należy ustawić na szachownicy osiem hetmanów tak, by się nawzajem nie szachowały (dwa hetmany szachują się nawzajem, gdy stoją w tej samej kolumnie lub w tym samym wierszu lub na tej samej linii skośnej). (a) Przykładowa konfiguracja. (b) Stan pośredni, dla którego heurystyczna funkcja kosztu  $h$  zwraca wartość 17. Pokazano także wartości  $h$  dla wszystkich następników tego stanu — wpisana w dane pole liczba oznacza wartość  $h$  dla stanu powstającego przez ustawienie hetmana na tym polu. Spośród możliwych następników osiem cechuje się najlepszą wartością  $h = 12$



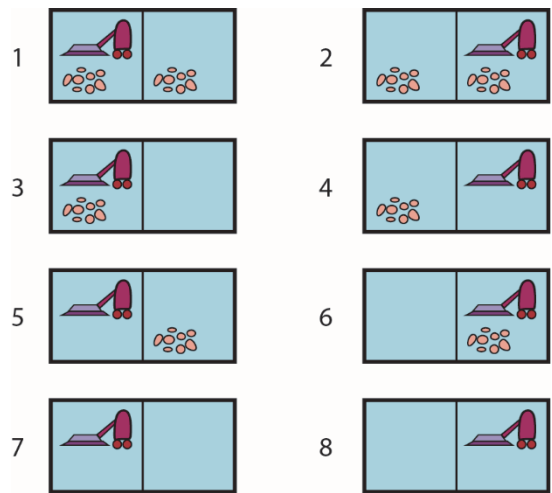
**RYСУNEK 4.3.** Oto dlaczego grzbiety górskie są kłopotliwe dla (zachłannego) wyszukiwania wspinaczkowego. Siatka stanów (oznaczonych ciemnymi elipsami) nakłada się na grzbiet wznoszący się od lewej do prawej strony, tworząc sekwencję lokalnych maksimów, które nie są ze sobą bezpośrednio połączone. Z każdego lokalnego maksimum wszystkie dostępne akcje skutkują ruchem w dół. Topologie takie jak ta są powszechne w przestrzeniach stanów niskiego wymiaru, takich jak punkty na płaszczyźnie. W przestrzeniach o setkach czy tysiącach wymiarów ten intuicyjny obraz traci jednak rację bytu, ponieważ istnieje zwykle kilka wymiarów umożliwiających ucieczkę z grzbietu lub płaskowyżu.



**RYСУNEK 4.4.** Krok przykładowego algorytmu genetycznego, rozwiązującego problem ośmiu hetmanów; reprezentacja osobnika odzwierciedla numery wierszy, w których znajdują się poszczególne hetmany. Osobniki tworzące początkową populację (a) oceniane są — w części (b) — przez funkcję przystosowania na podstawie liczby kolidujących par hetmanów. Najlepiej oceniane osobniki zostają wybrane (c) do generowania potomstwa (d). Potomstwo staje się następnie obiektem losowych mutacji (e)

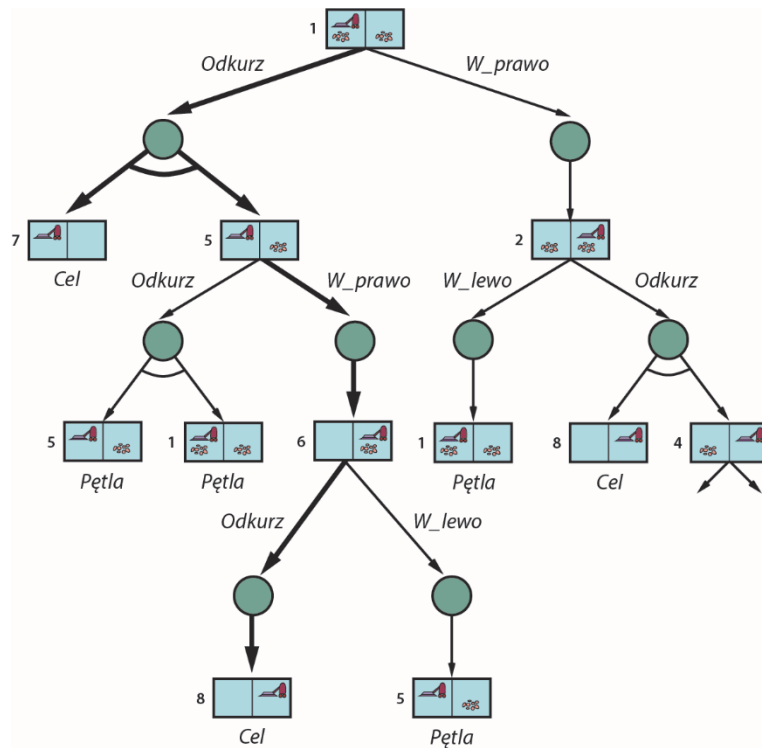


**RYСУNEK 4.5.** Stany problemu ośmiu hetmanów odpowiadające pierwszej parze rodziców z części (c) rysunku 4.4 i pierwszemu potomstwu z części (d). Kolumny w kolorze zielonym tracone są w procesie krzyżowania, zachowywane są kolumny w kolorze czerwonym. W reprezentacjach osobników na rysunku 4.4 cyfra 1 oznacza najniższy wiersz szachownicy, cyfra 8 najwyższy

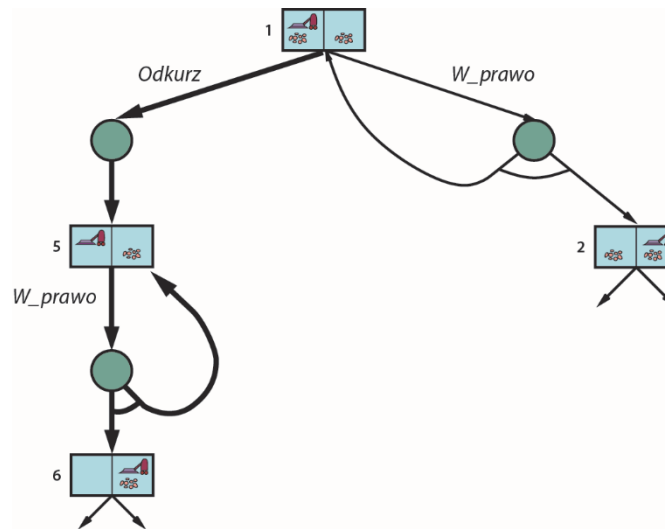


RYSUNEK 4.6. Osiem możliwych stanów agenta odkurzacza, stany 7 i 8 są stanami docelowymi

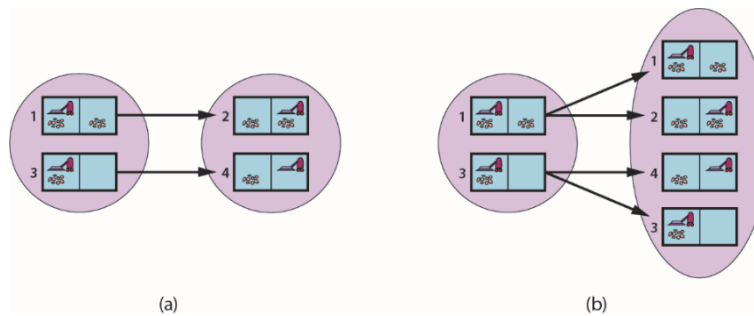
---



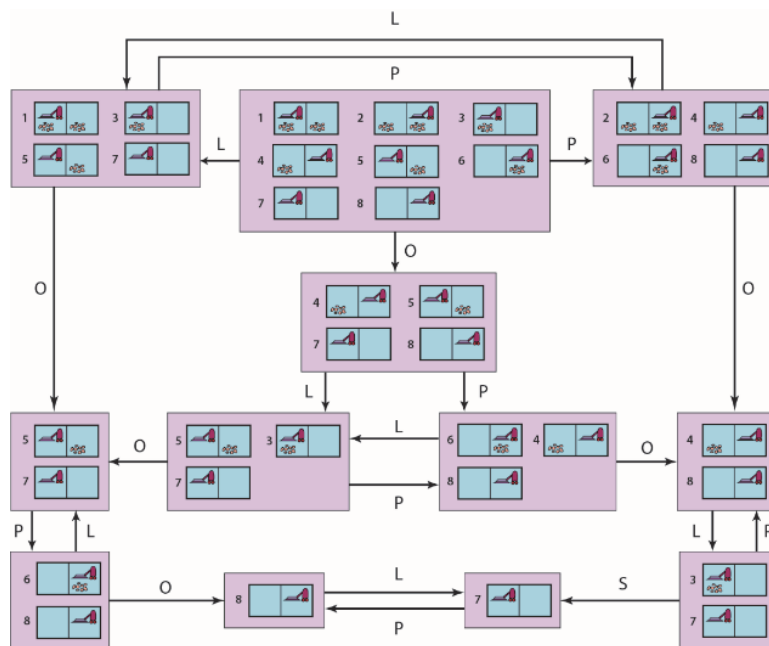
**RYСУNEK 4.7.** Dwa pierwsze poziomy drzewa wyszukiwawczego kapryśnego agenta-odkurzacza. Węzły ilustrujące stany to węzły OR reprezentujące wybór akcji przez agenta. Kółka oznaczają węzły AND, reprezentujące konieczność uwzględnienia wszystkich możliwych stanów wynikowych, co symbolizowane jest przez wychodzące z tych węzłów gałęzie. Gałęzie prowadzące do stanów docelowych wyróżnione są przez pogrubienie.



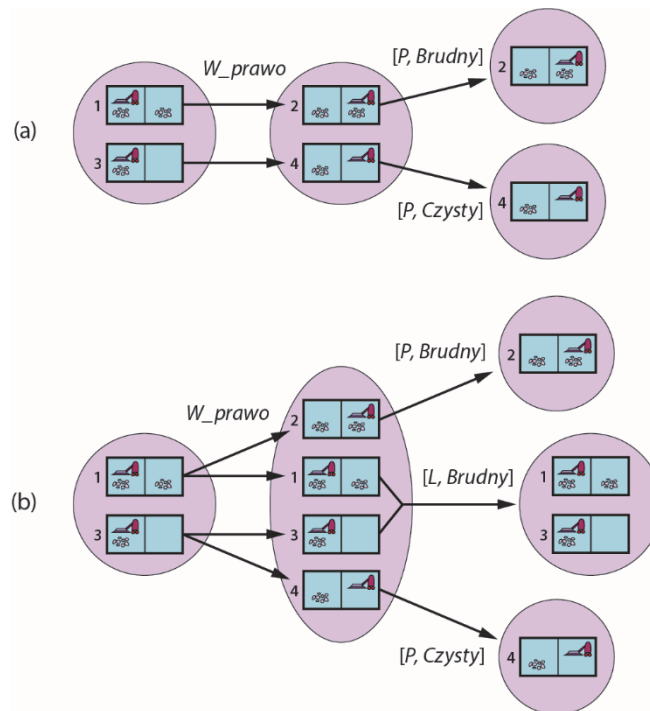
RYSUNEK 4.8. Fragment drzewa wyszukiwawczego agenta-odkurzacza z upośredzonym poruszaniem się, pokazaliśmy jawnie niektóre cykle. Wszystkie rozwiązania są cyklicznymi planami, nie można bowiem wymusić niezawodnego przemieszczania agenta



RYSUNEK 4.9. (a) Przewidywanie nowego stanu przekonań dla bezczujnikowego problemu agenta-odkurzacza, dla deterministycznej akcji  $W\_prawo$ . (b) Przewidywanie dla tego samego stanu przekonań dla niedeterministycznego („kapryśnego”) odkurzacza

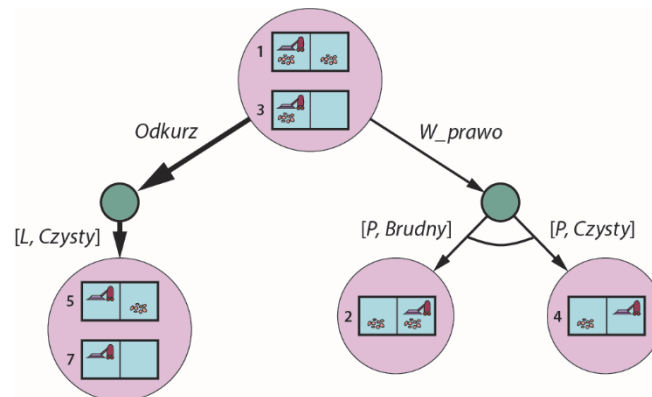


**RYСУNEK 4.10.** Osiągalna porcja przestrzeni stanów przekonań dla bezczujnikowego, deterministycznego agenta-odkurzacza. Każdy prostokątny panel odpowiada pojedynczemu stanowi przekonań. W każdym punkcie dla agenta określony jest pewien stan przekonań, lecz nieznanym jest dla niego bieżący stan fizyczny. Początkowy stan przekonań — kompletna niewiedza — reprezentowany jest przez najwyższy panel obejmujący wszystkich 8 możliwych stanów fizycznych. Litery oznaczają akcje podejmowane przez agenta: L — *W\_lewo*, P — *W\_prawo*, O — *Odkurz*

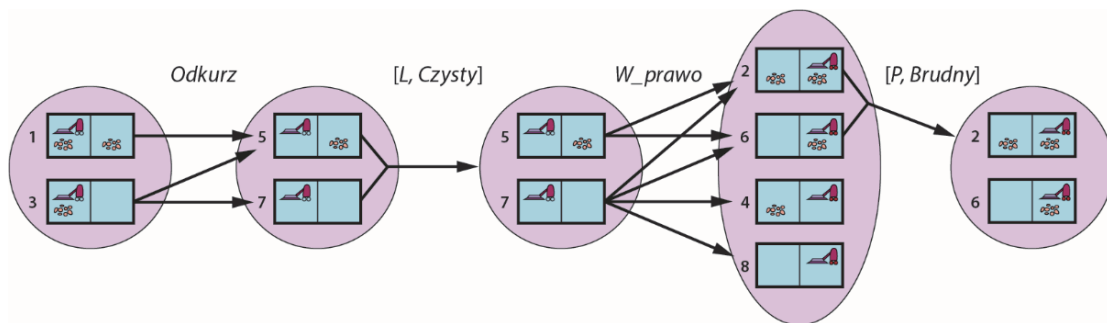


**RYSUNEK 4.11.** Dwa przykłady przejść w lokalnym środowisku agenta-odkurzacza. (a) W deterministycznym świecie efektem wykonania akcji  $W\_prawo$  w początkowym stanie przekonań jest nowy przewidywany stan przekonań x dwoma możliwymi stanami fizycznymi; w tych stanach możliwe są percepty  $[P, Czysty]$  i  $[P, Brudny]$ , prowadzące do dwóch singletonowych stanów przekonań. (b) W przypadku „kapryśnego” odkurzacza wykonanie akcji  $W\_prawo$  w początkowym stanie przekonań skutkuje nowym stanem przekonań o czterech składowych, w których możliwymi perceptami są  $[L, Brudny]$ ,  $[P, Brudny]$  i  $[P, Czysty]$ .

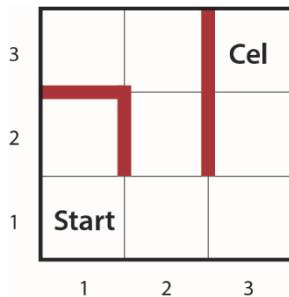




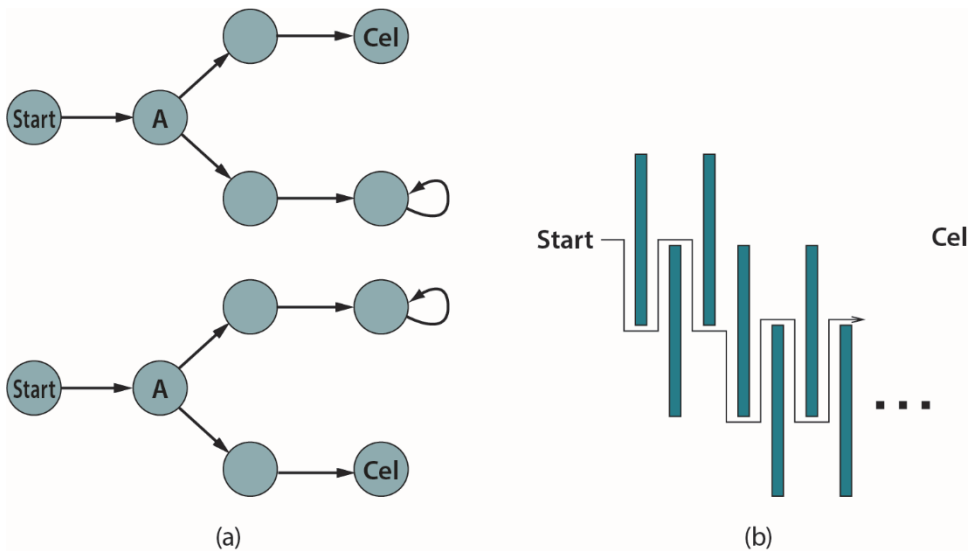
RYSUNEK 4.12. Pierwszy poziom drzewa wyszukiwawczego dla problemu agenta-odkurzacza z lokalną percepcją. Pierwszą akcją rozwiązania jest *Odkurz*



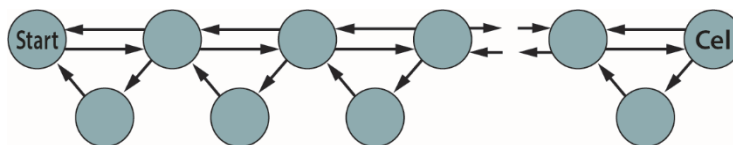
RYSUNEK 4.13. Dwa cykle „przewidywanie-aktualizacja” stanów przekonań agenta-odkurzacza z lokalną percepcją, zmagającego się przejawami inwencji przedszkolaków



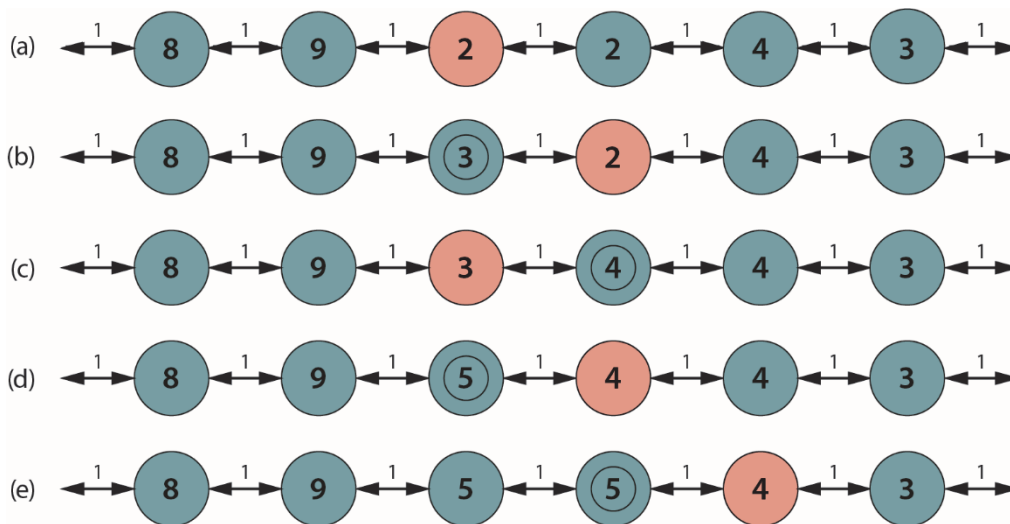
**RYSUNEK 4.15.** Prosty labirynt. Agent znajduje się na polu *Start* i musi dojść do pola *Cel*



**RYSUNEK 4.16.** (a) Dwie przesłuzzenie stanów, które mogą zaprowadzić agenta wyszukiwania online w ślepy zaułek — każdy agent wpadnie w taką pułapkę przynajmniej w jednej z nich. (b) Środowisko dwuwymiarowe, które może poprowadzić agenta wyszukiwania online dowolnie nieefektywną ścieżką: jakiegokolwiek wyboru nie dokonałby agent, przeciwnik blokuje mu drogę, stawiając (kolejną) długą, cienką ścianę, przez co ścieżka agenta wydłuża się znacznie w stosunku do najlepszej możliwej ścieżki



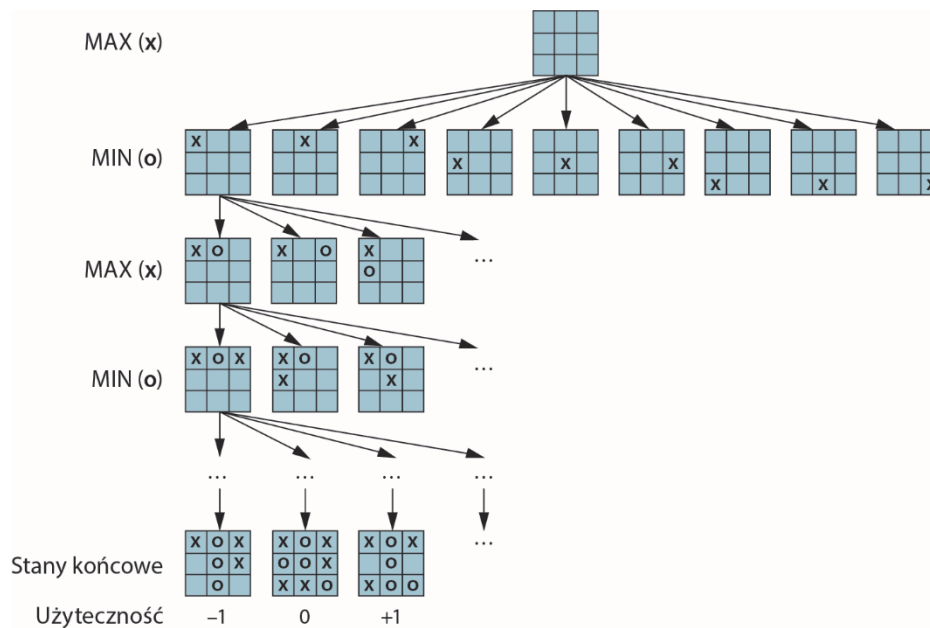
RYSUNEK 4.17. Przykład środowiska, w którym osiągnięcie celu za pomocą spaceru losowego wymaga wykładniczo wielu kroków



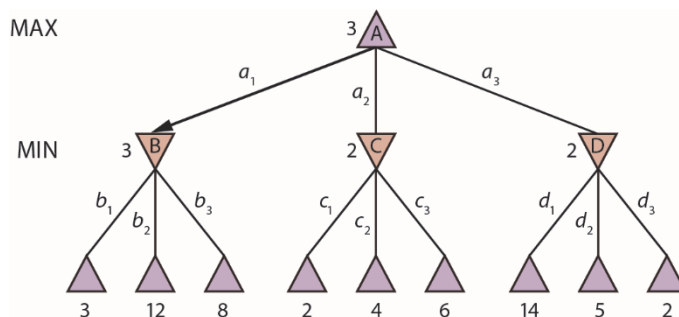
RYSUNEK 4.18. Pięć iteracji LRTA\* w jednowymiarowej przestrzeni stanów. Każdy stan etykietowany jest aktualną wartością  $H(s)$  — bieżącym oszacowaniem kosztu osiągnięcia celu — a koszt każdego łącza (akcji) wynosi 1. Kolor czerwony oznacza bieżącą lokalizację agenta, a aktualizowane w danej iteracji oszacowanie kosztu wyróżnione jest dodatkowym okręgiem.

## ROZDZIAŁ 5

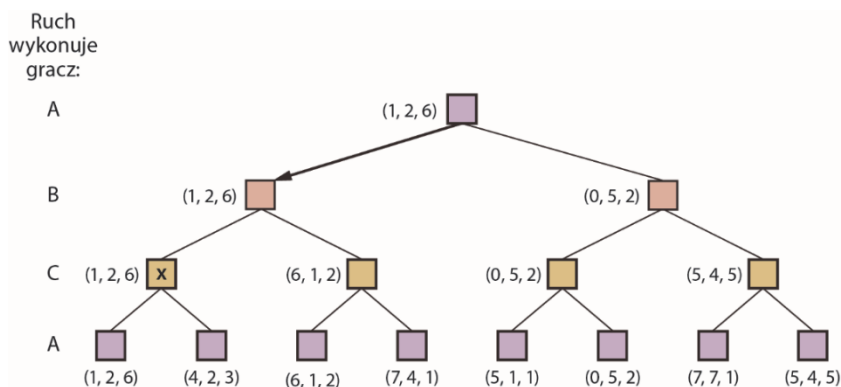
# WYSZUKIWANIE ANTAGONISTYCZNE I GRY



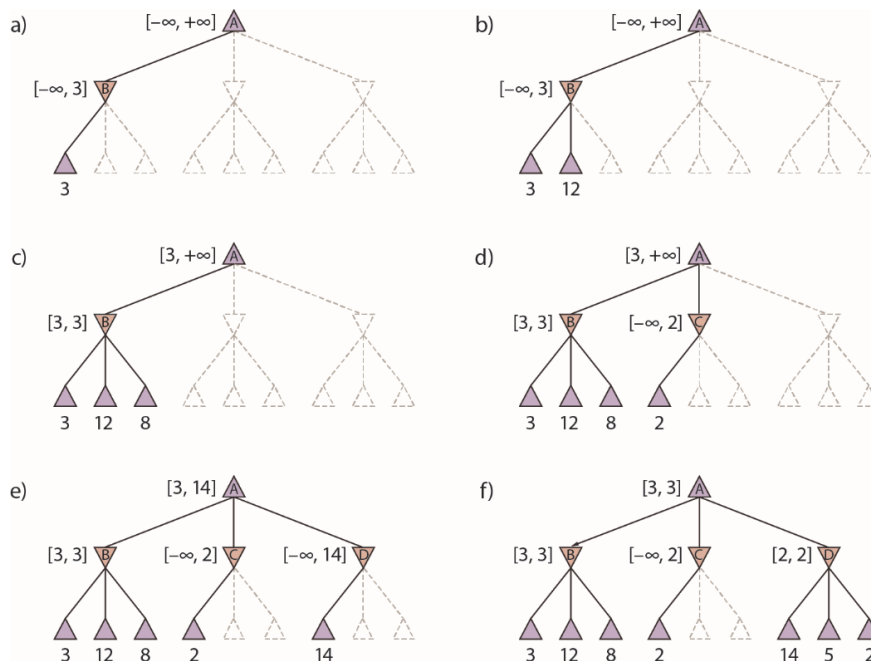
**RYSUNEK 5.1.** Drzewo gry (częściowe) dla gry w kółko i krzyżyk. Szczytowy węzeł reprezentuje stan końcowy, MAX wykonuje pierwszy ruch umieszczając krzyżyk w jednym z pustych kwadratów. Widzimy część drzewa odzwierciedlającą naprzemienne ruchy MIN-a (O) i MAX-a (X), aż osiągnięty zostanie któryś ze stanów końcowych, z wartością użyteczności określonej przez reguły gry



**RYСУNEK 5.2.** Drzewo gry, w której wykonuje się tylko dwa posunięcia. Węzły  $\triangle$  to węzły gracza MAX, węzły  $\nabla$  reprezentują ruchy gracza MIN. Węzły końcowe etykietowane są wartościami użyteczności z punktu widzenia MAX-a. Dla pozostałych węzłów podano także wartości minimaksowe. Najlepszym ruchem MAX-a na poziomie korzenia jest  $a_1$ , ponieważ prowadzi do stanu z *największą* wartością minimaksu; najlepszą odpowiedzią MIN-a jest ruch  $b_1$  jako prowadzący do stanu o *najmniejszej* wartości minimaksu

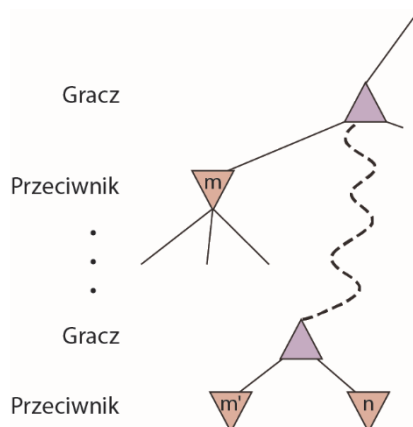


**RYСУNEK 5.3.** Trzy pierwsze posunięcia gry z udziałem trzech graczy  $(A, B, C)$ . Każdy węzeł etykietowany jest wektorem odzwierciedlającym jego użyteczność z punktu widzenia poszczególnych graczy. Wektor najlepszego ruchu zostaje ostatecznie zwrócony do korzenia drzewa

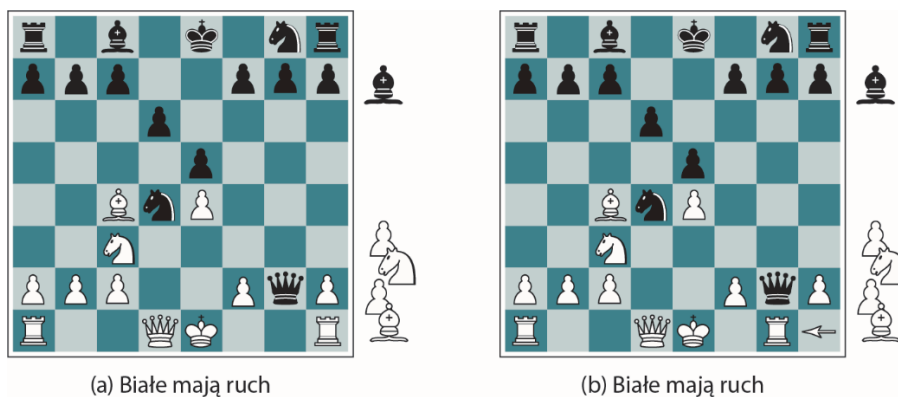


**RYSUNEK 5.4.** Etapy obliczania optymalnej decyzji dla drzewa gry z rysunku 5.2, dla każdego węzła podaliśmy zakres jego możliwych wartości.

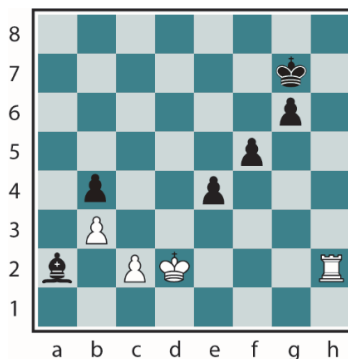
- Pierwszy liść poniżej  $B$  ma wartość 3, zatem  $B$ , jako węzeł gracza MIN, musi mieć wartość nie większą od 3.
- Drugi liść poniżej  $B$  ma wartość 12, która nie wpływa na zakres możliwych wartości węzła  $B$  — gracz MIN, jako gracz optymalny, nie będzie brał tego ruchu pod uwagę.
- Trzeci liść poniżej  $B$  ma wartość 8; znamy już wartości wszystkich następników węzła  $B$ , którego wartość zostaje tym samym *ustalona* jako 3. Wnioskujemy stąd, że wartość korzenia musi wynosić *co najmniej* 3 (korzeń jest węzłem gracza MAX).
- Pierwszy liść poniżej  $C$  ma wartość 2, więc  $C$  jako węzeł gracza MIN, musi mieć wartość nie większą od 2. Gracz MAX nigdy nie wybierze tego ruchu, bo węzeł  $B$  ma większą wartość 3, zatem *możemy wykluczyć z dalszej analizy wszystkie następniki węzła  $C$*  — to jest właśnie przykład przycinania alfa-beta.
- Pierwszy liść poniżej  $D$  ma wartość 14, więc  $D$  jako węzeł gracza MIN, musi mieć wartość nie większą od 14. Jest to najlepszy dotychczas wybór gracza MAX, musimy więc przeanalizować następniki węzła  $D$ . Znamy już wszystkie następniki korzenia: pierwszy ma wartość 3, drugi wartość nie większą niż 14; wnioskujemy stąd, że korzeń nie może mieć wartości większej niż 14.
- Drugi następnik węzła  $D$  ma wartość 5, wciąż musimy więc kontynuować eksplorację. Trzeci następnik węzła  $D$  ma wartość 2, co ustala wartość  $D$  na dokładnie 2. Skoro znamy już wartości wszystkich następników korzenia (3, nie więcej niż 2, 2) wnioskujemy, że wartością korzenia jest 3.



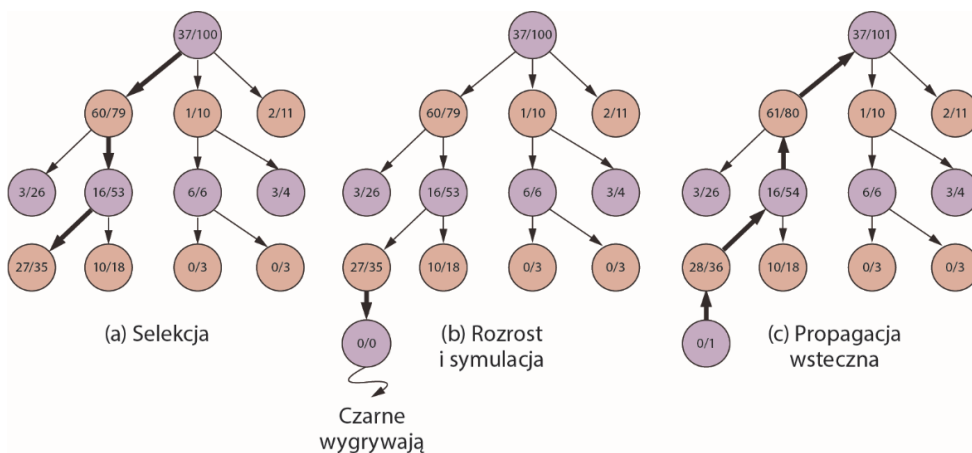
**RYSUNEK 5.5.** Ogólna zasada przycinania alfa-beta. Jeśli węzły  $m'$  lub  $m$  są lepsze dla gracza niż węzeł  $n$ , gracz nie wykonuje ruchu do węzła  $n$



**RYSUNEK 5.6.** Dwa stany gry różniące się jedynie pozycją wieży w prawym dolnym rogu. (a) Czarne mają przewagę materialną w sile dwóch pionków i jednego skoczka ( $2 \times 1 + 3 = 5$ ), co powinno być wystarczające do zwycięstwa. (b) Biała wieża bije czarnego hetmana i teraz białe zyskują przewagę materialną ( $9 - (2 \times 1 + 3) = 4$ ), co powinno wystarczyć do zwycięstwa

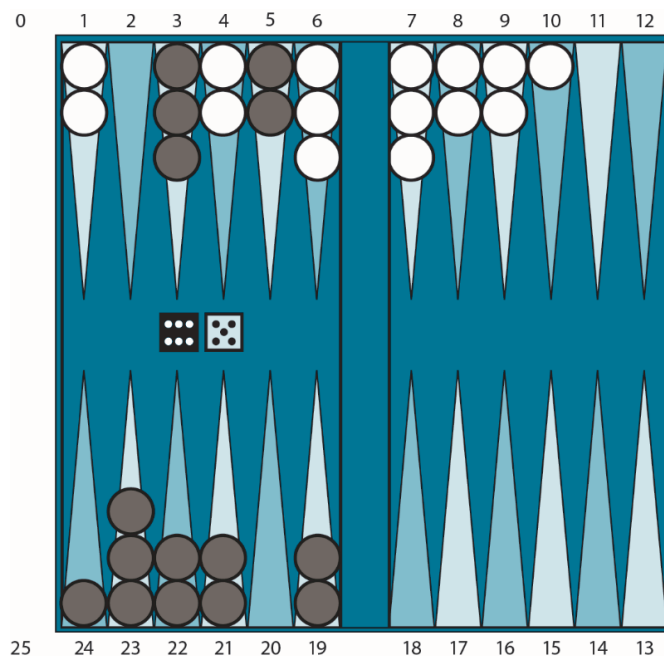


**RYСУNEK 5.7.** Czarne mają ruch, los ich gońca jest już przesądzony. Czarne próbują jednak zapobiegać zbiciu gońca, dwukrotnie szachując króla pionkami, prowokując go do zbitia tych pionków. Istnieje możliwość, że ruchy te odsuną perspektywę zbitia gońca poza horyzont dostrzegalności wyszukiwania i wspomniane posunięcia czarnych zostaną ocenione jako dobre.

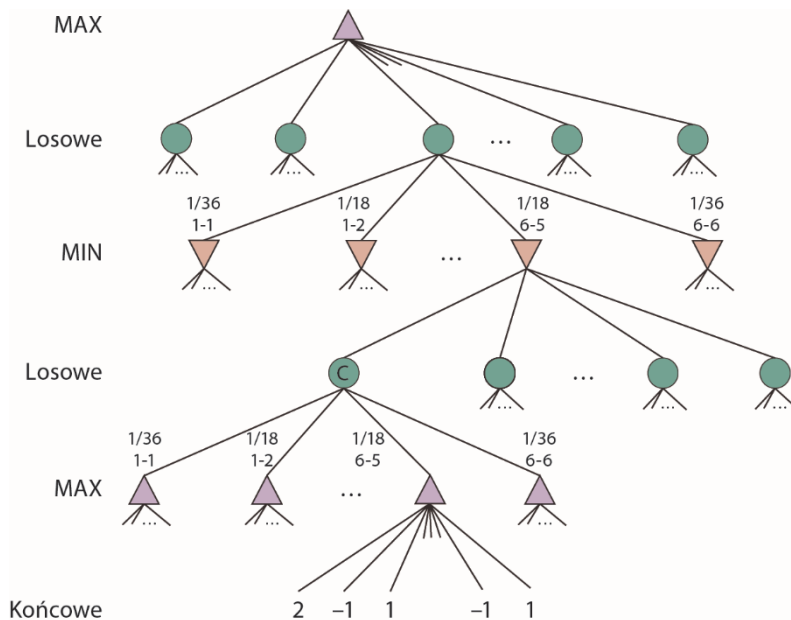


**RYСУNEK 5.8.** Pojedyncza iteracja wyboru ruchu w drzewiastym wyszukiwaniu Monte Carlo, przy użyciu metryki „górnej granicy przedziału ufności w odniesieniu do drzew” (UCT); rysunek pokazuje sytuację po wykonaniu 100 iteracji. W części (a) startujemy od korzenia i schodzimy w głąb drzewa do liścia o wskaźniku 27/35 (co oznacza 27 zwycięstw czarnych w 35 rozgrywkach). W części (b) pokazany jest elementarny krok rozbudowy drzewa: generujemy węzeł potomny wspomnianego liścia i przeprowadzamy symulację (rozgrywkę) dla tego nowego węzła — symulacja ta kończy się zwycięstwem czarnych. Część (c) ilustruje propagację wsteczną wyniku symulacji do korzenia drzewa

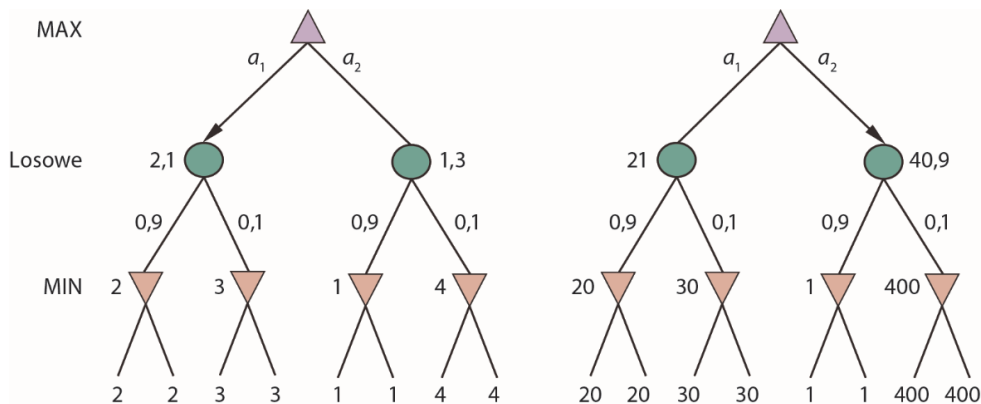




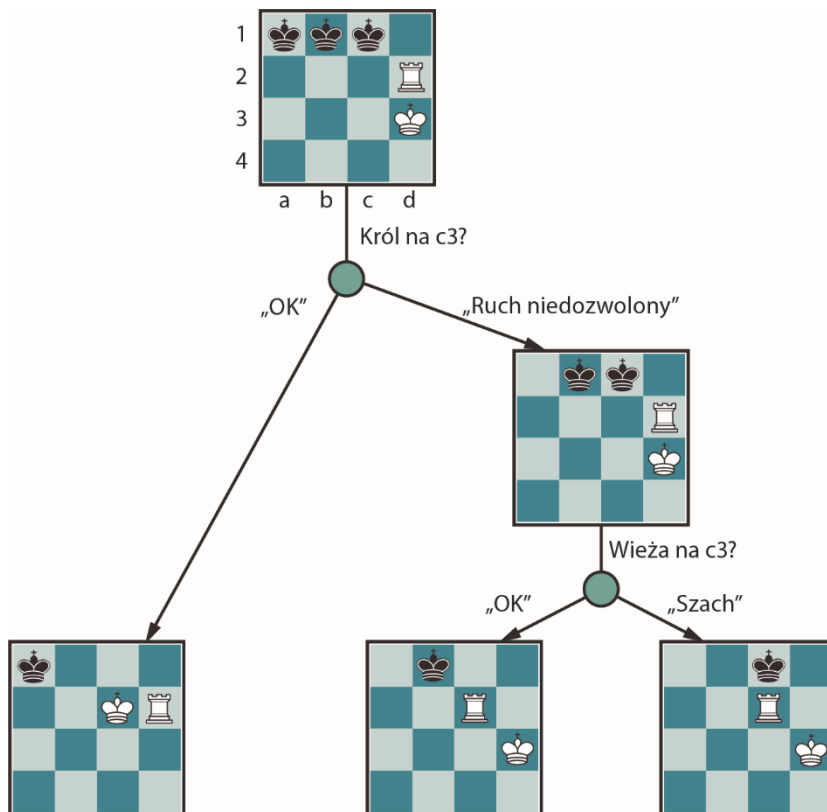
**RYСУNEK 5.9.** Typowa pozycja gry tryktrak. Celem gracza jest usunięcie z planszy wszystkich swoich pionków — wygrywa ten, który uczyni to jako pierwszy. Czarne pionki poruszają się zgodnie ze wskazówkami zegara w kierunku pola 25, białe poruszają się antyzegarowo w kierunku pola 0. Pionek może zająć dowolne pole, jeżeli jest ono puste lub znajduje się na nim tylko jeden pionek przeciwnika. Samotny pionek przeciwnika zostaje wtedy zbity; przeciwnik nie może kontynuować gry, dopóki jego pionek nie wróci na planszę po rzucie kostką. W przedstawionej na rysunku pozycji gracz Czarny wyrzucił 6 oraz 5 i musi wybrać jeden z czterech dozwolonych ruchów:  $(5 \rightarrow 11, 5 \rightarrow 10)$ ,  $(5 \rightarrow 11, 19 \rightarrow 24)$ ,  $(5 \rightarrow 10, 10 \rightarrow 16)$  lub  $(5 \rightarrow 11, 11 \rightarrow 16)$  — czyli jeden pionek przesuwany o 5 pozycji, drugi o 6



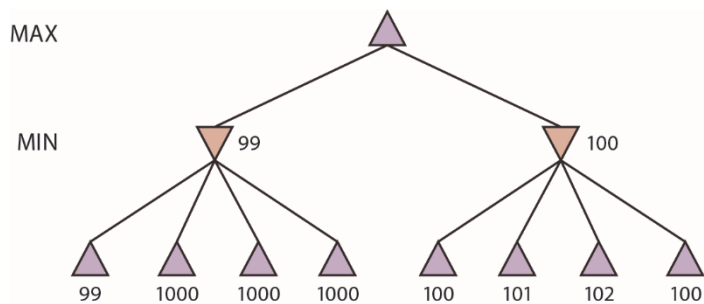
RYSUNEK 5.10. Schematyczne drzewo gry w tryktraka dla określonej pozycji



RYSUNEK 5.11. Nawet przy zachowaniu monotoniczności zmiana wartości funkcji ewaluacyjnej może skutkować zmianą decyzji o wyborze najlepszego ruchu



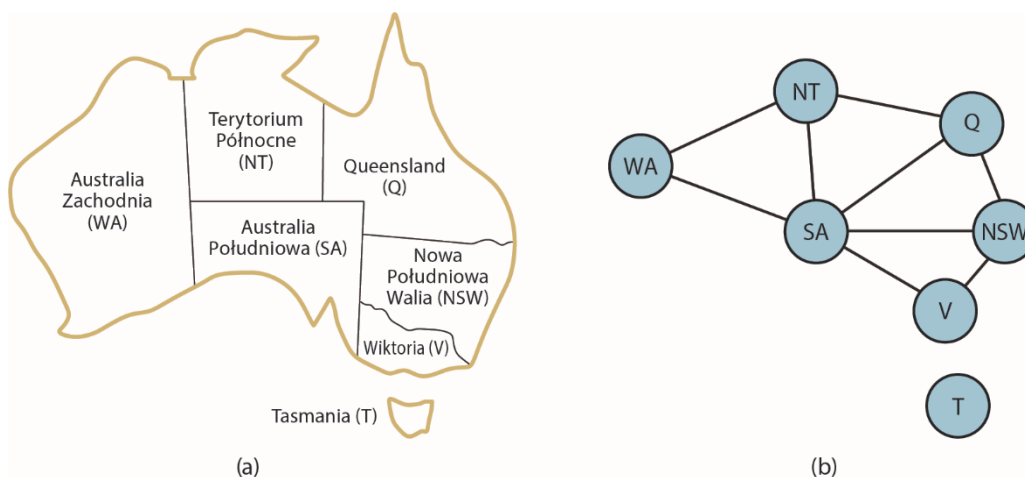
**RYSUNEK 5.12.** Część gwarantowanego mata dla końcówki KRK na fragmencie szachownicy. W początkowym stanie przekonania Białego czarny król może znajdować się na jednej z trzech pozycji; w wyniku kombinacji próbnych ruchów Białego jego stan przekonania zostaje zredukowany do jednej pozycji (doprowadzenie tej gry do mata pozostawiamy Czytelnikowi jako ćwiczenie)



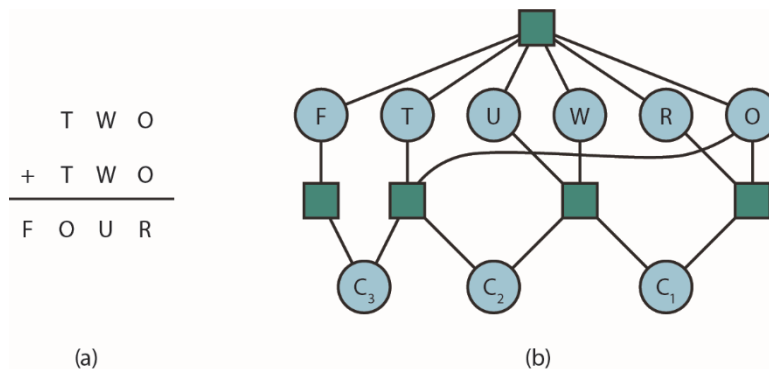
RYSUNEK 5.13. Drzewo gry, dla którego węzłów funkcja ewaluacyjna obarczona jest pewnym błędem

## ROZDZIAŁ 6

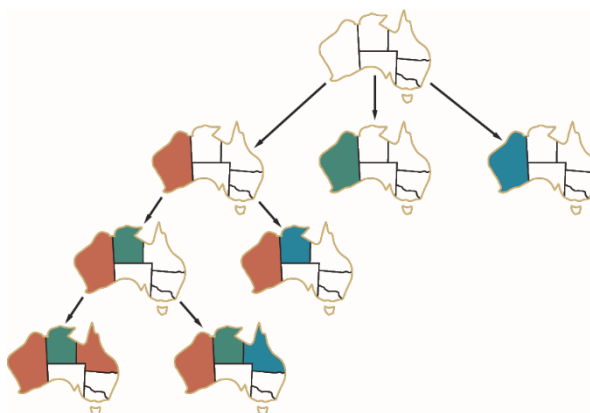
# PROBLEMY SPEŁNIANIA OGRANICZEŃ



**RYSUNEK 6.1.** (a) Główne regiony Australii. Pokolorowanie tej mapy w taki sposób, by graniczące ze sobą regiony miały różne kolory, może być uważane za instancję problemu spełniania ograniczeń (CSP). (b) Reprezentacja problemu pokolorowania mapy w postaci grafu ograniczeń



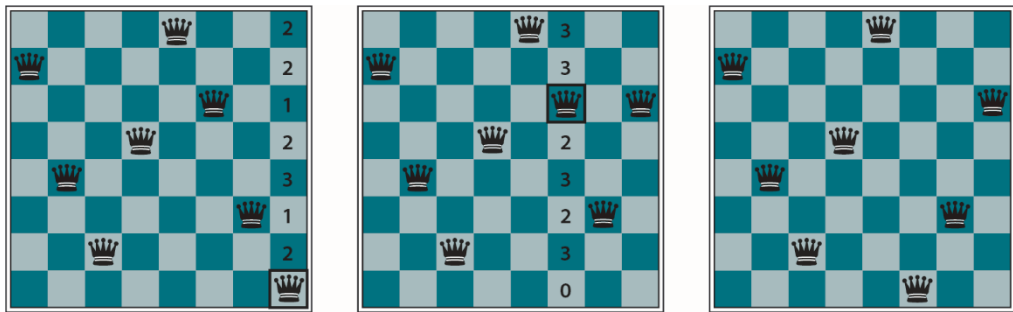
**RYSUNEK 6.2.** (a) Kryptarytm jako instancja problemu CSP. Każda litera odpowiada cyfrze dziesiętnej, a problem polega na znalezieniu takiego przyporządkowania cyfr poszczególnym literom, by powstało poprawne działanie arytmetyczne — z dodatkowym zastrzeżeniem, że najbardziej żadna z liczb nie może zawierać wiodących zer. (b) Hipergraf ograniczeń dla tego problemu; szczytowy kwadrat reprezentuje ograniczenia Alldiff, kwadraty na niższym poziomie reprezentują ograniczenia wynikające z reguł dodawania liczb w dziesiętnym systemie liczbowym. Zmienne pomocnicze  $C_1$ ,  $C_2$  i  $C_3$  reprezentują przeniesienia na wyższe pozycje (w kolejności od prawej do lewej)



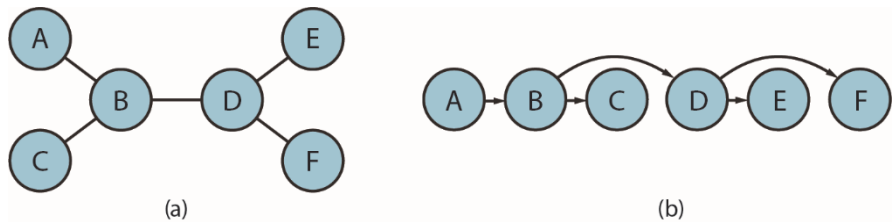
**RYSUNEK 6.4.** Część drzewa wyszukiwawczego dla problemu kolorowania mapy z rysunku 6.1

	WA	NT	Q	NSW	V	SA	T
Początkowo							
Po przypisaniu WA = czerwony							
Po przypisaniu Q = zielony							
Po przypisaniu V = niebieski							

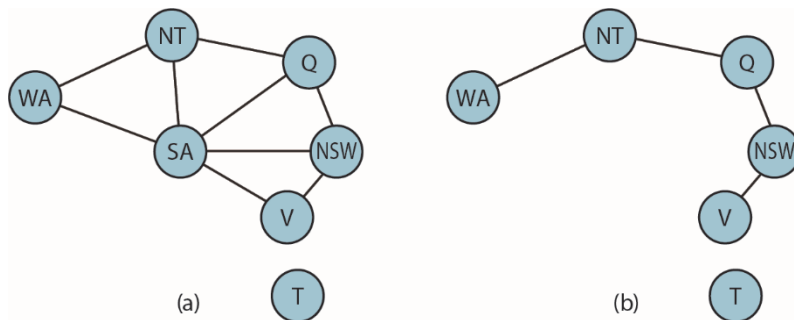
**RYSUNEK 6.5.** Postęp wyszukiwania ze sprawdzaniem w przód rozwiązującego problem kolorowania mapy z rysunku 6.1. Po dokonaniu początkowego przypisania *WA = czerwony* wyszukiwanie w przód usuwa wartość *czerwony* z domen zmiennych sąsiadujących z *WA*. Po przypisaniu *Q = zielony* wartość *zielony* zostaje usunięta z domen zmiennych *NT*, *SA* i *NSW*. Po przypisaniu *V = niebieski* wartość *niebieski* zostaje usunięta z domen zmiennych *NSW* i *SA*, co doprowadza do opróżnienia domeny zmiennej *SA*



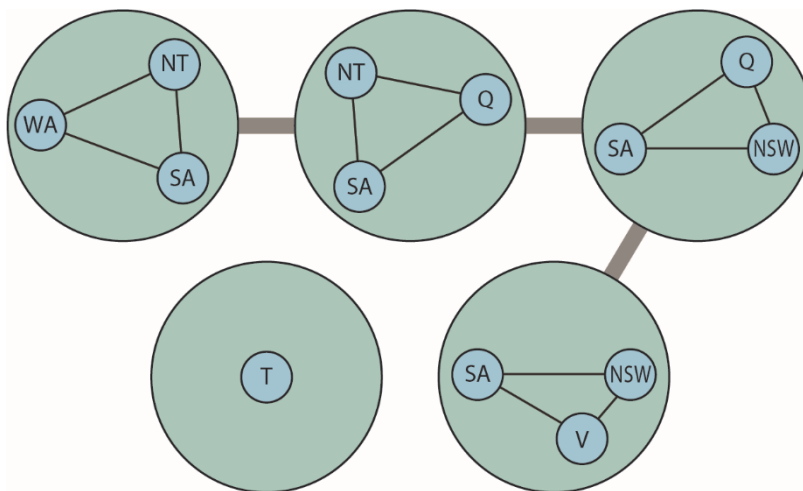
**RYSUNEK 6.6.** Rozwiązanie problemu 8 hetmanów w dwóch krokach, przy użyciu heurystyki minimalnych konfliktów. W każdym kroku wybierany jest hetman do przemieszczenia w jego własnej kolumnie, dla każdej z możliwych pozycji pokazano liczbę konfliktów, czyli liczbę innych hetmanów szachujących hetmana postawionego na teże pozycji. Algorytm przesuwa hetmana na pole najmniej skonfliktowane. Ewentualne remisy (w tym przypadku dwie pozycje o jednym konflikcie) rozstrzygane są w sposób losowy



**RYSUNEK 6.7.** (a) Graf ograniczeń dla problemu CSP o strukturze drzewiastej. (b) Liniowe uporządkowanie zmiennych spójne z drzewem o korzeniu *A*. Takie porządkowanie nosi nazwę **sortowania topologicznego** zmiennych



**RYSUNEK 6.8.** (a) Oryginalny graf ograniczeń dla problemu kolorowania mapy z rysunku 6.1. (b) Po usunięciu wierzchołka *SA* graf staje się lasem złożonym z dwóch drzew

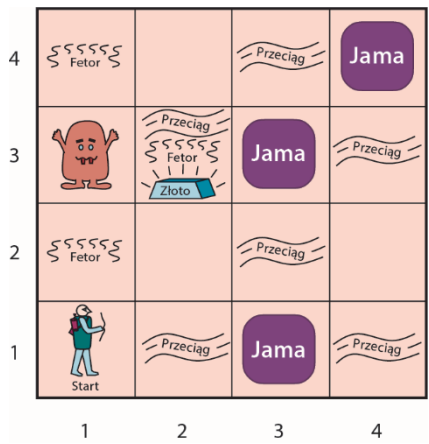


**RYSUNEK 6.9.** Dekompozycja drzewiasta grafu ograniczeń z rysunku 6.8 (a)

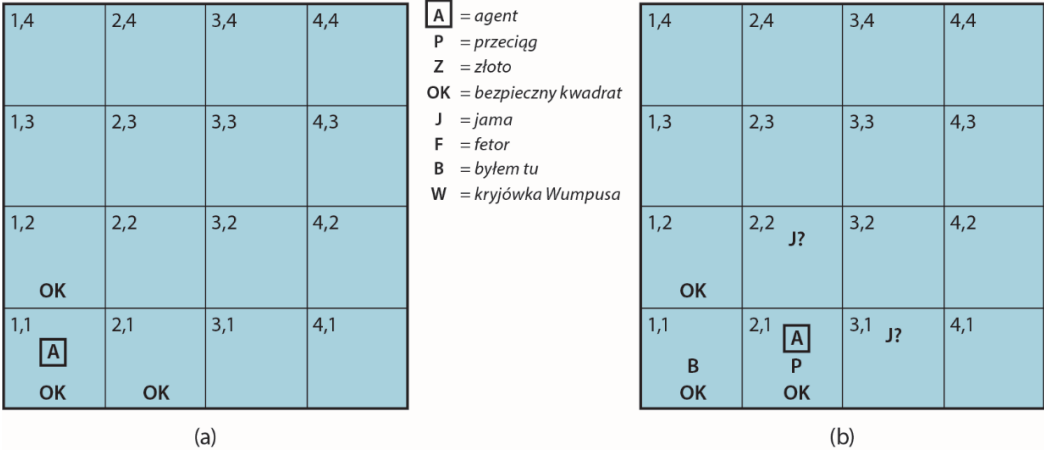


ROZDZIAŁ 7

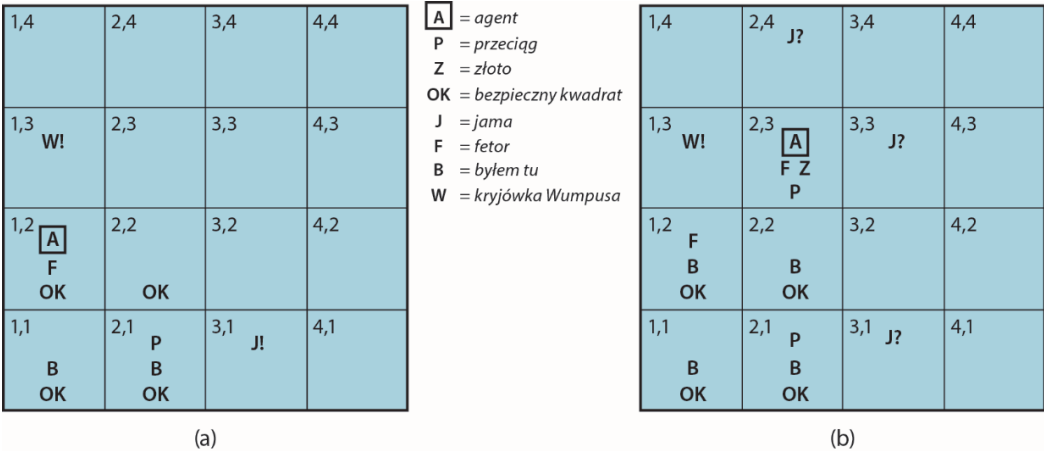
LOGICZNI AGENCI



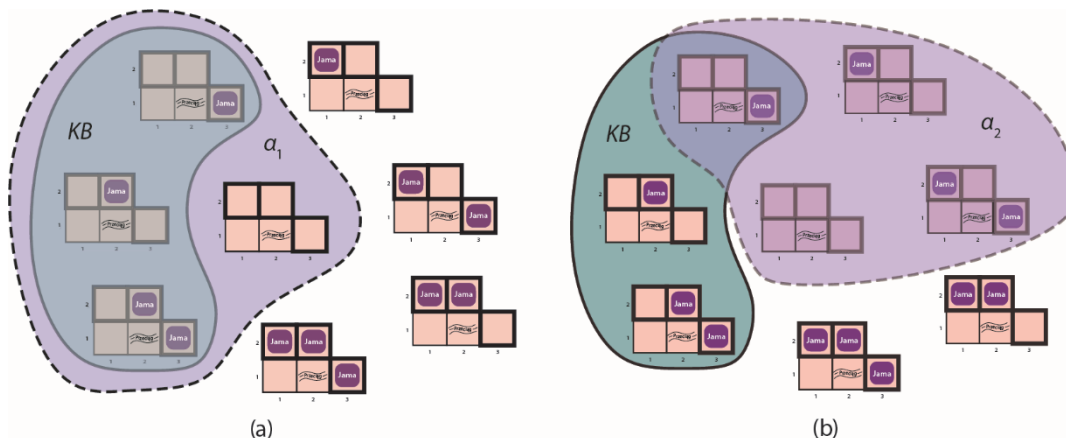
RYSUNEK 7.1. Typowa konfiguracja świata Wumpusa. Początkowo agent znajduje się w lewym dolnym kwadracie i zwrócony jest w prawą stronę



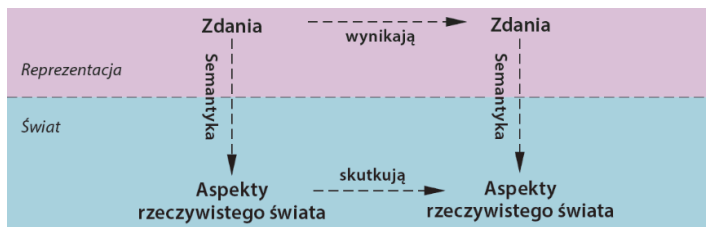
**RYСУNEK 7.2.** Pierwszy krok agenta w świecie Wumpusa. (a) Sytuacja początkowa po odebraniu „pustego” percepcju [Nic, Nic, Nic, Nic, Nic]. (b) Sytuacja po przemieszczeniu agenta na kwadrat (2, 1) i odebraniu percepcju [Nic, Przeciąg, Nic, Nic, Nic]



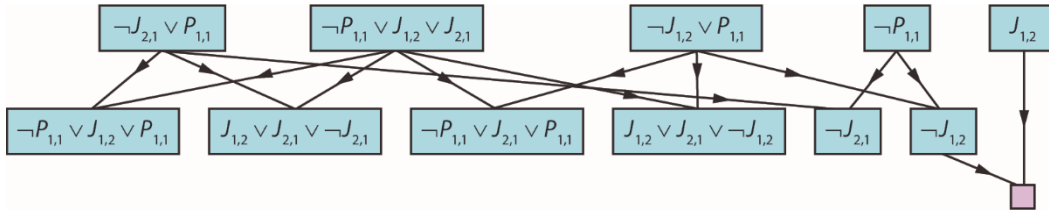
**RYСУNEK 7.3.** Dwa kolejne etapy życia agenta. (a) Po przemieszczeniu agenta na (1, 1), następnie na (1, 2) i odebraniu percepcju [Fetor, Nic, Nic, Nic, Nic]. (b) Po przemieszczeniu agenta na (2, 2), następnie na (2, 3) i odebraniu percepcju [Fetor, Przeciąg, Blask, Nic, Nic]



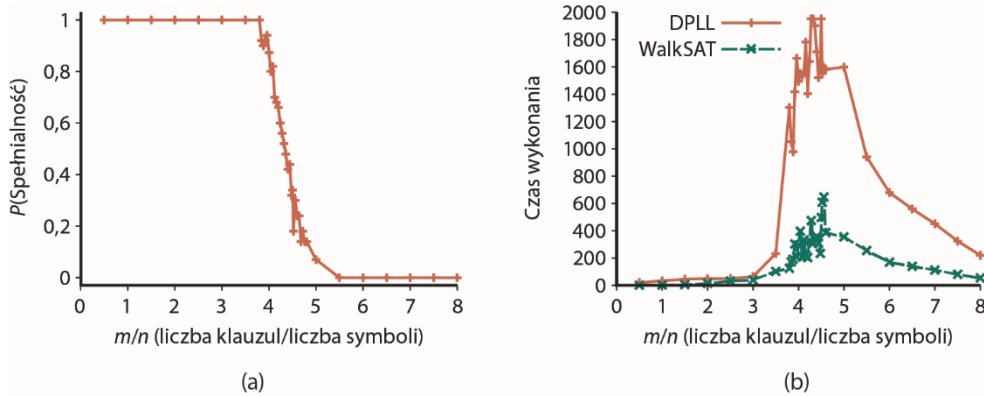
**RYSUNEK 7.4.** Możliwe modele obecności jam w kwadratach (1, 2), (2, 2) i (3, 1). Baza wiedzy odpowiadająca „pustemu” perceptorowi w kwadracie (1, 1) i przeciągowi w kwadracie (2, 1) zakreślona jest linią ciągłą. (a) Linia kropkowana ukazuje modele zdania  $\alpha_1$  (nie ma jamy w (1, 2)). (b) Linia kropkowana ukazuje modele zdania  $\alpha_2$  (nie ma jamy w (2, 2))



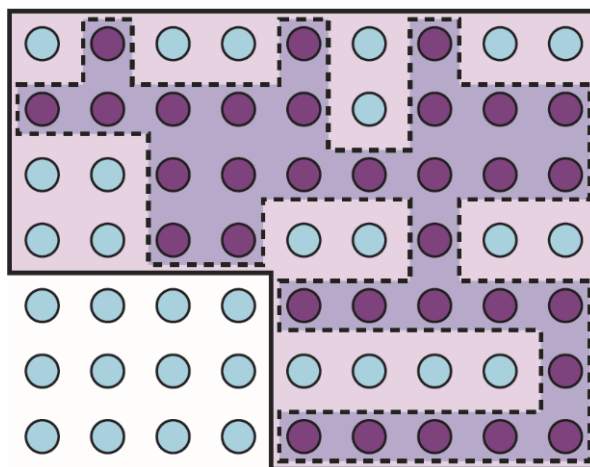
**RYSUNEK 7.5.** Zdania stanowią fizyczne konfiguracje agenta, a wnioskowanie jest procesem konstruowania nowych konfiguracji fizycznych na podstawie istniejących. Celem logicznego wnioskowania powinno być zapewnienie, że nowe konfiguracje odzwierciedlają te aspekty świata rzeczywistego, które wynikają z aspektów odpowiadających istniejącym konfiguracjom.



**RYSUNEK 7.9.** Częściowe zastosowanie algorytmu PL-RESOLUTION do prostego wnioskowania w świecie Wumpusa, w celu udowodnienia prawdziwości  $\neg J_{1,2}$ . Każda z czterech klauzul w górnym rzędzie sparowana zostaje z pozostałymi trzema, a zastosowanie zasady rezolucji skutkuje klauzulami widocznymi w dolnym rzędzie. Zwróćmy uwagę, że kombinacja trzeciej i czwartej klauzuli z górnego rzędu daje klauzulę  $\neg J_{1,2}$ , która redukuje się z klauzulą  $J_{1,2}$ , dając klauzulę pustą — co oznacza, że oryginalne zdanie zostało udowodnione.



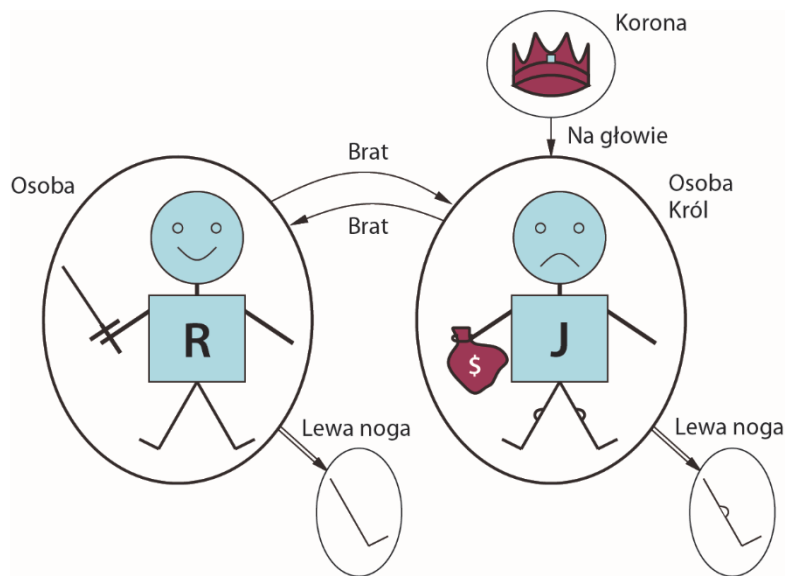
**RYSUNEK 7.11 WYKRES** prawdopodobieństwa spełnialności losowego zdania 3-CNF z  $n=50$  symbolami w zależności od ilorazu  $m/n$  ( $m$  jest liczbą klauzul). (b) Wykres mediany czasu wykonania (mierzonego liczbą iteracji) dla algorytmów DPLL i WALKSAT na zbiorze losowych zdań 3-CNF. Dla najtrudniejszych problemów wartość  $m/n$  plasuje się w okolicy 4,3



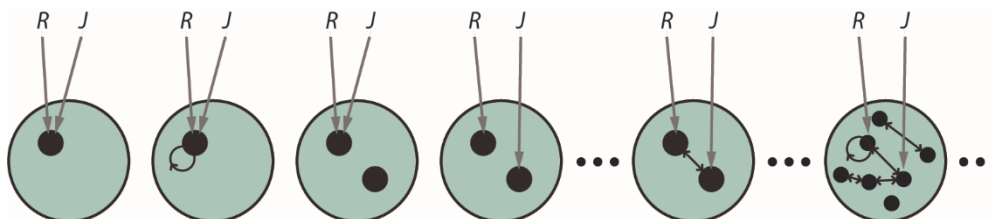
**RYSUNEK 7.12.** Ilustracja stanu przekonań 1-CNF (linia ciągła) jako reprezentowalnej, zachowawczej aproksymacji rzeczywistego stanu przekonań (zaczienione regiony ograniczone przerywaną linią). Każdy możliwy świat reprezentowany jest przez kółko, te na szarym tle spójne są ze wszystkimi perceptorami

## ROZDZIAŁ 8

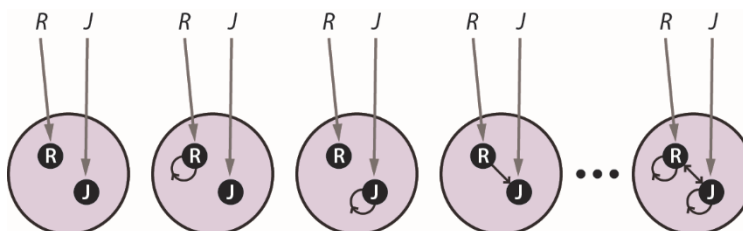
# LOGIKA PIERWSZEGO RZĘDU



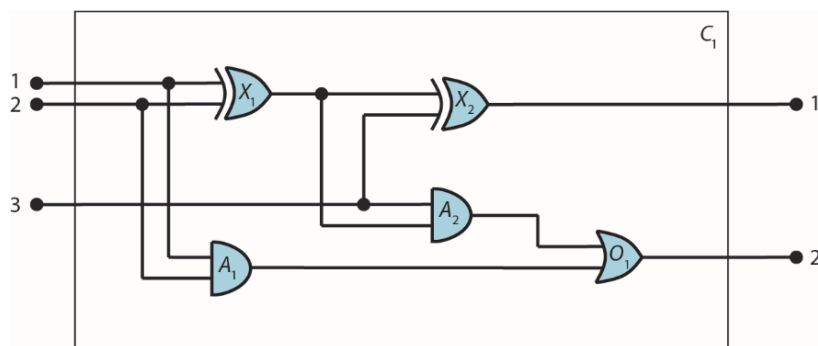
**RYSUNEK 8.1.** Przykładowy model zawierający pięć obiektów, dwie relacje binarne („brat” i „na głowie”), trzy relacje unarne („osoba”, „król” i „korona”) i jedną funkcję unarną („lewa noga”).



**RYSUNEK 8.3.** Niektóre elementy zbioru wszystkich możliwych modeli języka wykorzystującego dwa symbole stałe —  $R$  i  $J$  — i jeden symbol predykatowy. Interpretacje symboli stałych zaznaczone są szarymi strzałkami, w każdym modelu obiekty powiązane relacją połączone są strzałką



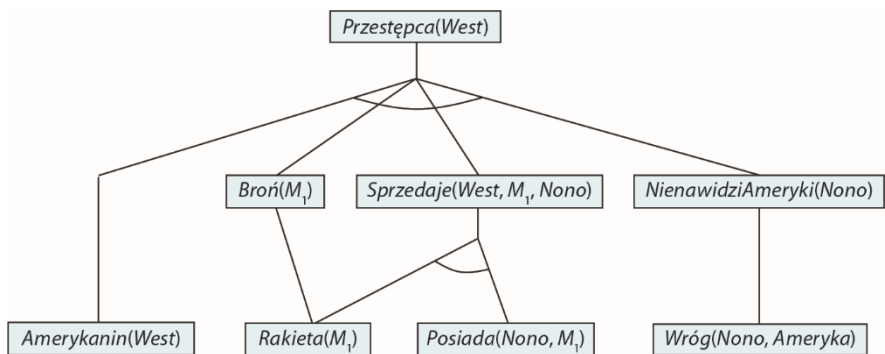
**RYSUNEK 8.4.** Kilka elementów zbioru wszystkich modeli dla języka z dwoma stałymi symbolami  $R$  i  $J$  oraz jednym symbolem relacyjnym (predykatowym) w semantyce bazodanowej. Interpretacja symboli stałych jest ustalona — każdemu odpowiada inny obiekt



**RYSUNEK 8.5.** Układ cyfrowy  $C_1$  pełniący rolę jednobitowego sumatora z przeniesieniem. Na wejścia 1 i 2 podaje się sumowane bity, na wejście 3 bit przeniesienia z poprzedniego sumatora. Na wyjściu 1 pojawia się bit sumy, na wyjściu 2 przeniesienie dla następnego sumatora. Układ zawiera dwie bramki XOR, dwie bramki AND i jedną bramkę OR

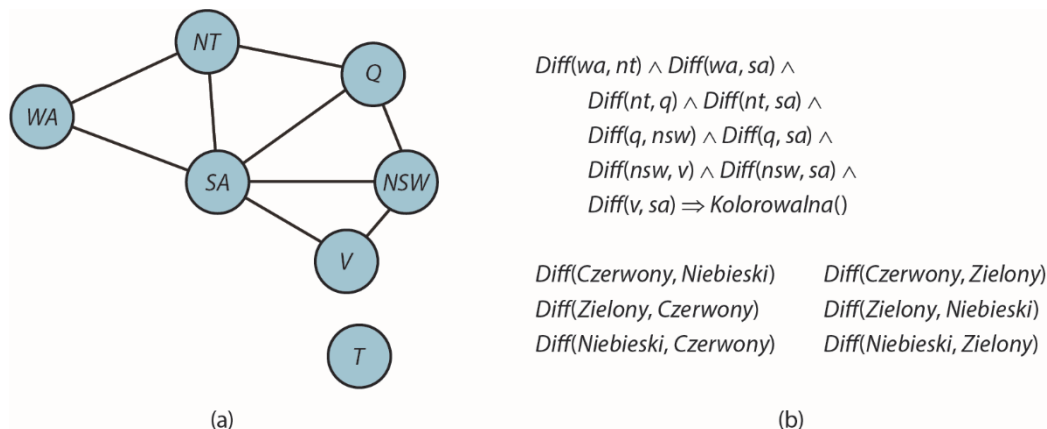
## ROZDZIAŁ 9

# WNIOSKOWANIE W LOGICE PIERWSZEGO RZĘDU

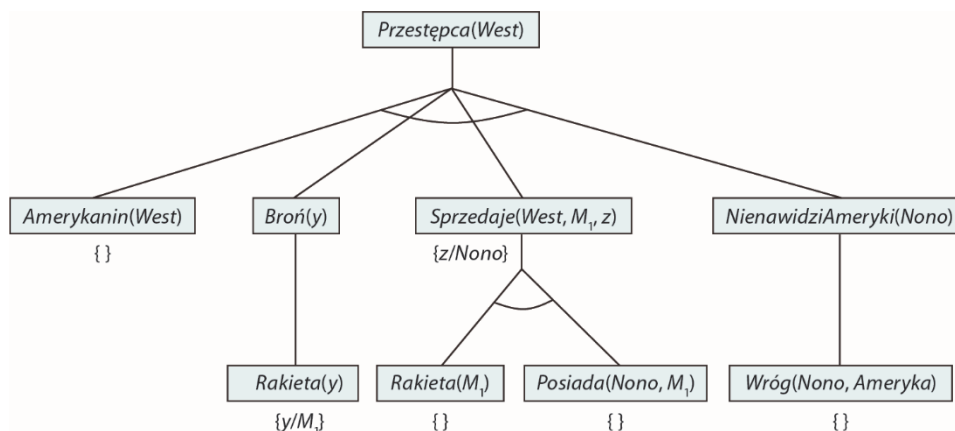


**RYSUNEK 9.2.** Drzewo dowodowe generowane przez łańcuchowanie progresywne w przykładzie z nielegalnym handlem bronią. Początkowe fakty pojawiają się na najniższym poziomie, fakty wywnioskowane w pierwszej iteracji plasują się na poziomie pośrednim, fakty wywnioskowane w drugiej iteracji — na poziomie najwyższym

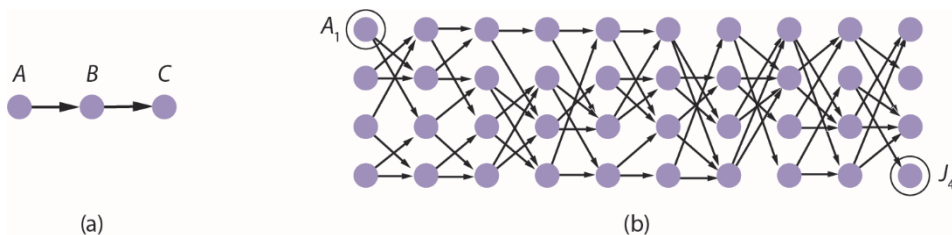




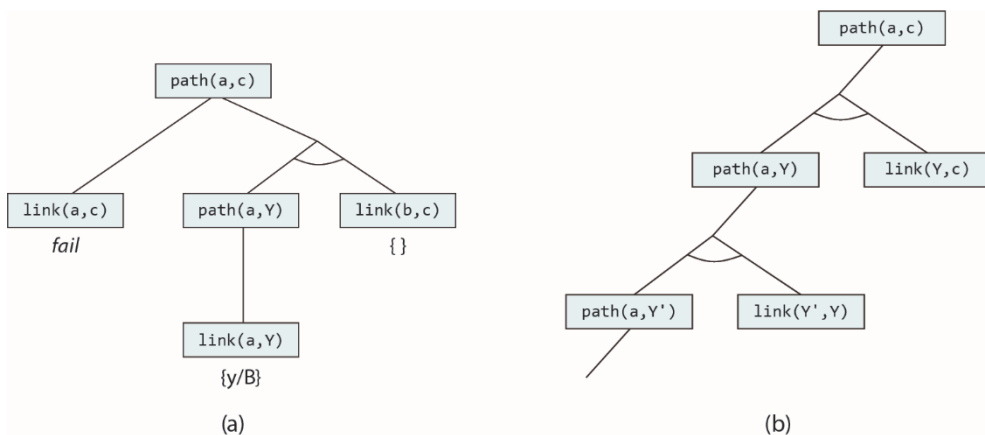
**RYСУNEK 9.3.** (a) Graf ograniczeń dla kolorowania mapy Australii. (b) Problem CSP kolorowania mapy wyrażony w postaci pojedynczej klauzuli definitywnej. Każdy region mapy reprezentowany jest przez zmienną, której wartością może być jedna ze stałych *Czerwony*, *Zielony* lub *Niebieski* (które deklarowane są jako różne — *Diff*)



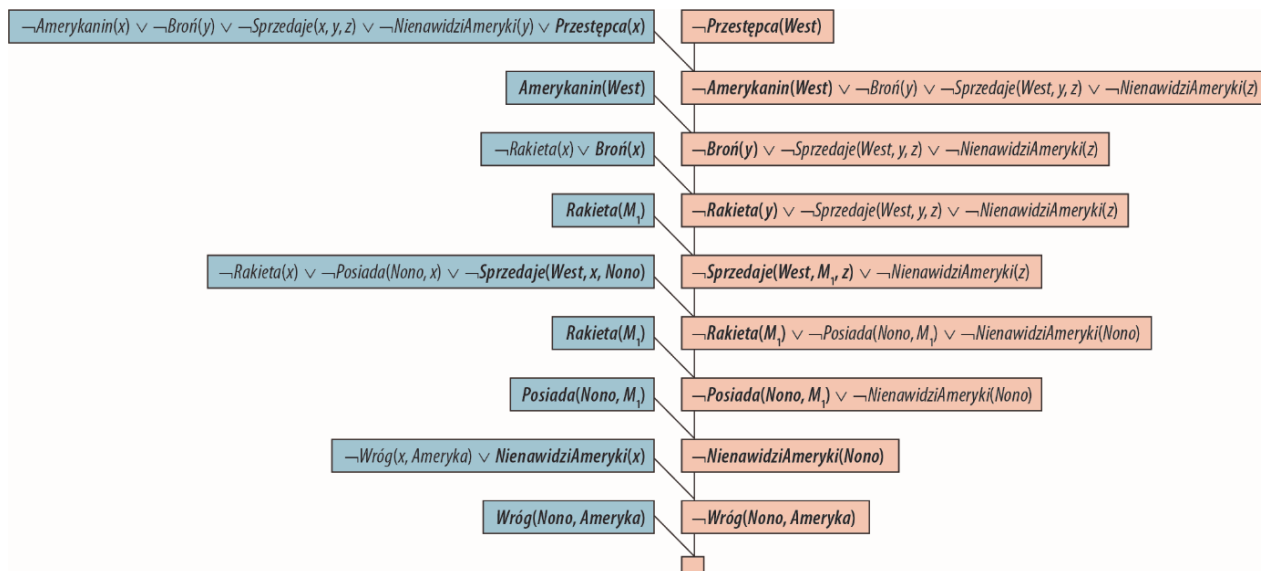
**RYСУNEK 9.4.** Drzewo dowodowe konstruowane przez algorytm łańcuchowania regresywnego w celu udowodnienia, że pułkownik West jest przestępcą. Drzewo to należy czytać od dołu do góry, od lewej do prawej. By udowodnić zdanie *Przestępcą(West)*, musimy udowodnić każdy z czterech koniunktów potomnych. Niektóre zapisane są już w bazie wiedzy, inne wymagają dalszego łańcuchowania. Wiązania dla każdej pomyślnej unifikacji pokazane są w pobliżu odnośnych celów. Zauważmy, że gdy tylko jeden z podcelów w koniunkcji zostanie udowodniony, związane z nim podstawienie stosowane jest do kolejnych podcelów. Kiedy więc funkcja FOL-BC-ASK przechodzi do ostatniego koniunktu w oryginalnym zdaniu *NienawidziAmeryki(z)*, zmienna *z* zostaje związana wartością *Nono*



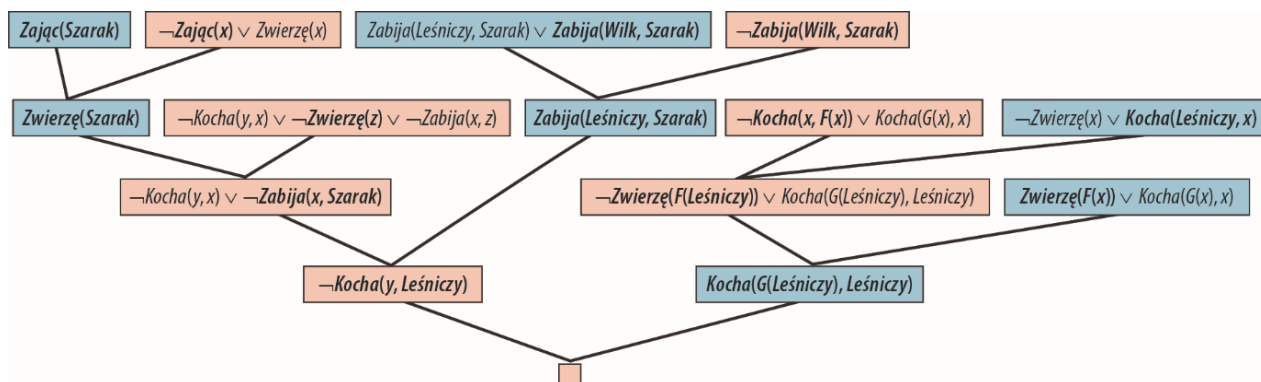
**RYSUNEK 9.5.** (a) Znajdowanie ścieżki od  $A$  do  $C$  może spowodować, że program w Prologu wpadnie w nieskończoną pętlę. (b) Graf, w którym każdy wierzchołek połączony jest z dwoma losowo wybranymi następnikami. Znalezienie ścieżki od  $A_1$  do  $J_4$  wymaga przeprowadzenia 877 wnioskowań



**RYSUNEK 9.6.** (a) Dowód istnienia ścieżki od  $A$  do  $C$ . (b) Nieskończone drzewo dowodowe generowane w wyniku „niewłaściwej” kolejności klauzul

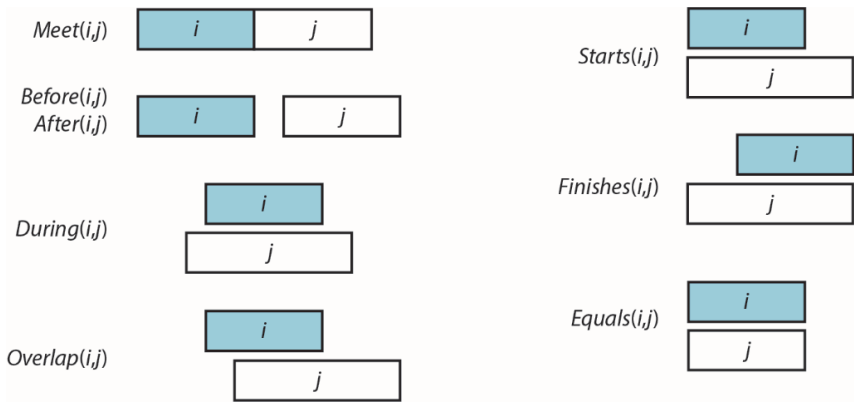


**RYСУNEK 9.7.** Dowód przez rezolucję, że pułkownik West jest przestępcą. W każdym kroku rezolucji unifikujące się literały wyróżnione są przez pogrubienie, a klauzule zawierające pozytywny literał znajdują się na niebieskim tle

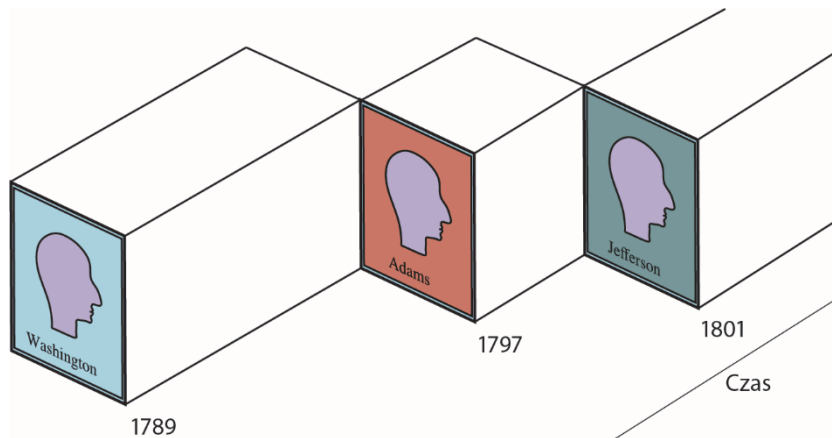


**RYСУNEK 9.8.** Dowód przez rezolucję, że to wilk zabił zająca. Zauważmy zastosowanie faktoryzacji przy wyprowadzaniu klauzuli  $Kocha(G(Leśniczy), Leśniczy)$ ; zwróćmy także uwagę, że unifikacja  $Kocha(x, F(x))$  i  $Kocha(G(x), x)$  może udać się dopiero po uprzedniej standaryzacji zmiennych

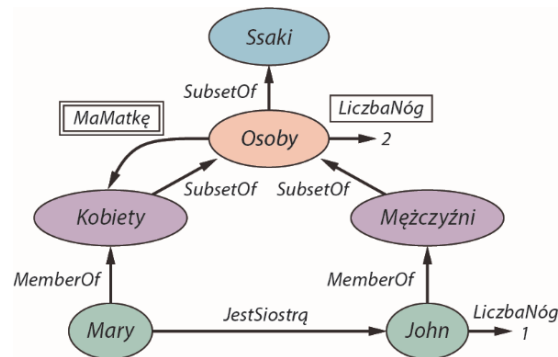
# REPREZENTACJA WIEDZY



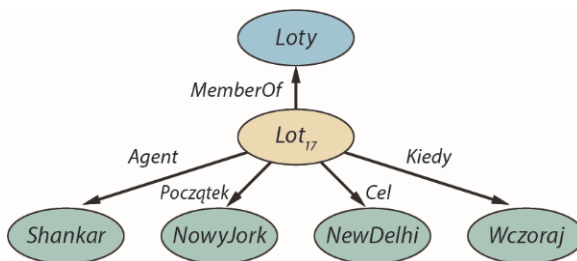
RYSUNEK 10.2. Predykaty interwałów czasowych



RYSUNEK 10.3. Schematyczne przedstawienie obiektu Prezydent(USA) u początków historii USA



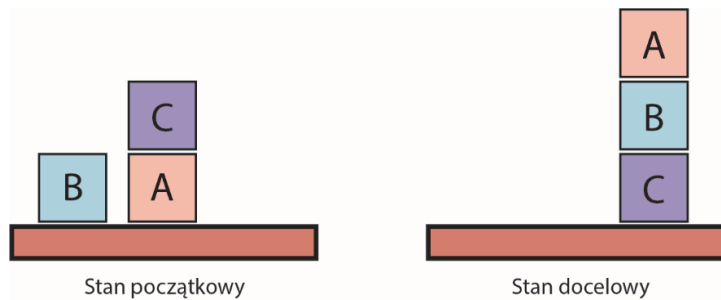
RYSUNEK 10.4. Sieć semantyczna z czterema obiektami (*John*, *Mary*, 1 i 2) i czterema kategoriami. Relacje reprezentowane są przez etykietowane łącza



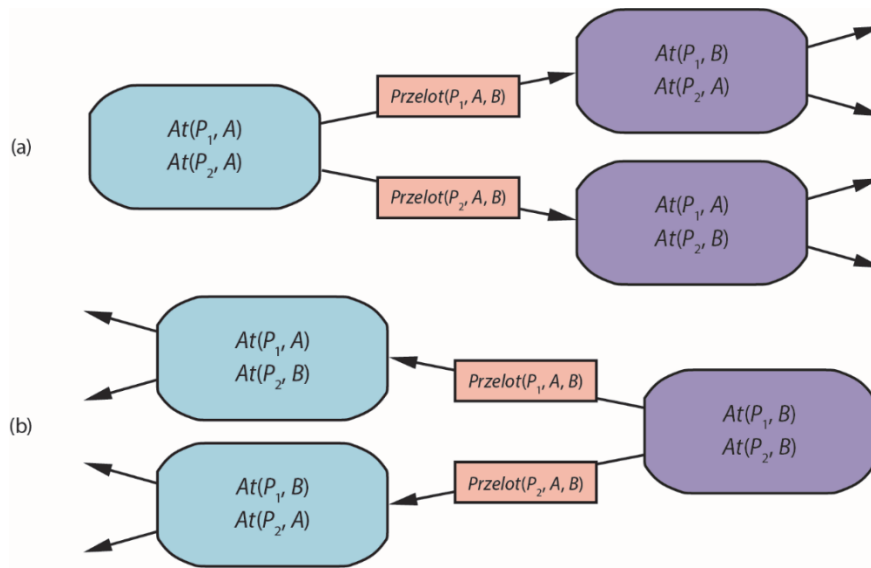
RYSUNEK 10.5. Fragment sieci semantycznej ukazujący reprezentację asercji *Lot(Shankar, NowyJork, NewDelhi, Wczoraj)*

## ROZDZIAŁ 11

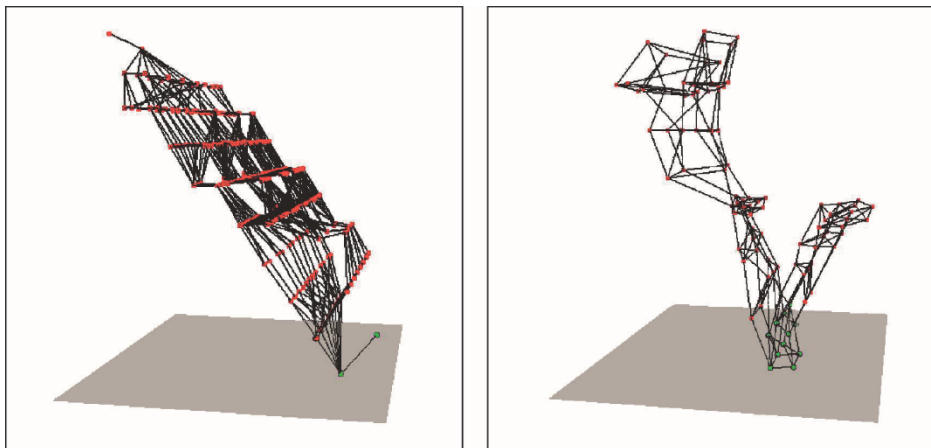
# AUTOMATYCZNE PLANOWANIE



RYSUNEK 11.1. Diagram świata blokowego dla planu z listingu 11.3

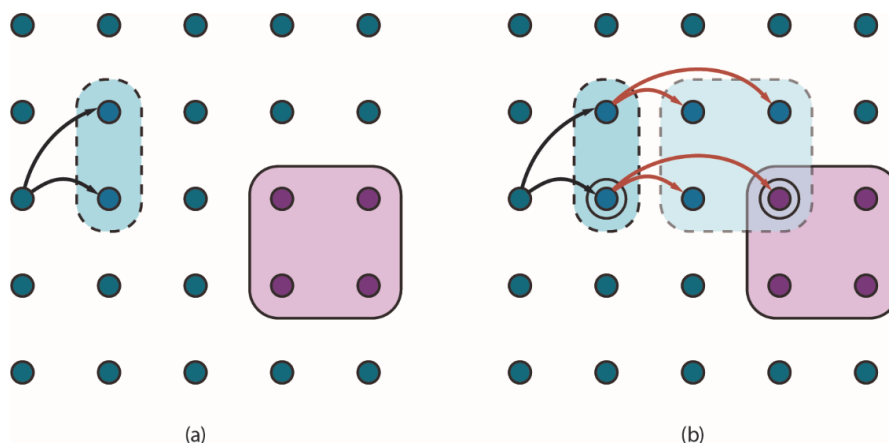


**RYSUNEK 11.2.** Dwa podejścia do wyszukiwania w planowaniu: (a) Wyszukiwanie w przód (progresywne) przestrzeni stanów podstawowych, rozpoczynane w stanie początkowym i wykorzystujące akcje problemu do znalezienia jednego ze stanów docelowych. (b) Wyszukiwanie wstecz (regresywne) w zbiorze opisów stanów, rozpoczynane w stanie docelowym i wykorzystujące odwrócenie akcji w celu dotarcia do stanu początkowego

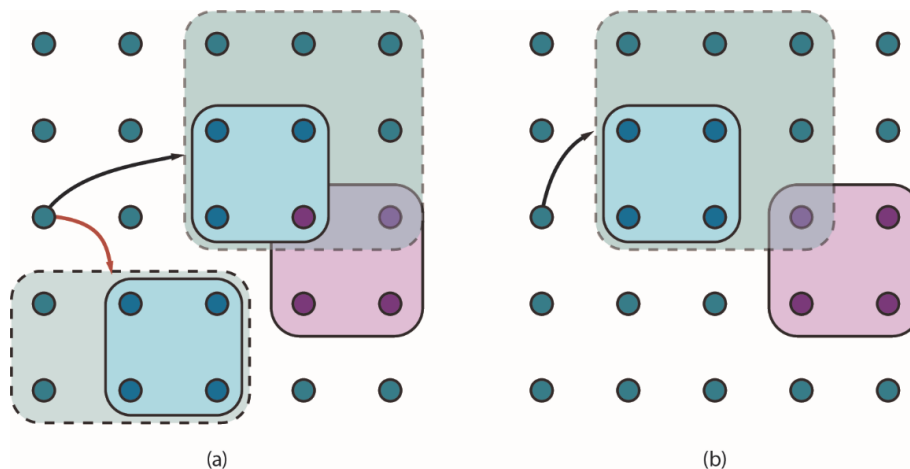


**RYSUNEK 11.3.** Dwie przestrzenie stanów problemu planowania z heurystyką ignorowania list usuwania. Wysokość punktu nad dolną płaszczyzną odzwierciedla wartość funkcji heurystycznej dla stanu reprezentowanego przez ten punkt; punkty leżące na wspomnianej płaszczyźnie reprezentują więc stany docelowe. Nie występują tu minima lokalne, więc wyszukiwanie jest nieskomplikowane. Rysunek zaczerpnięty z (Hoffmann, 2005)

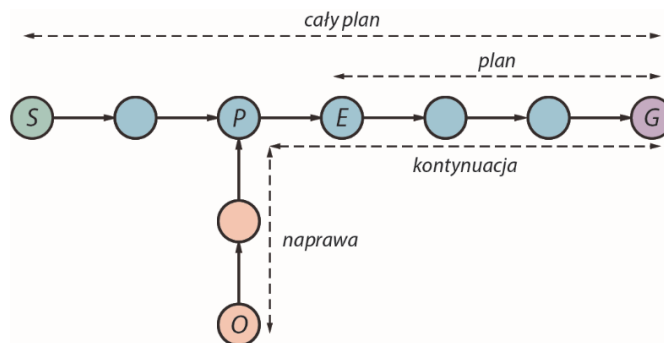




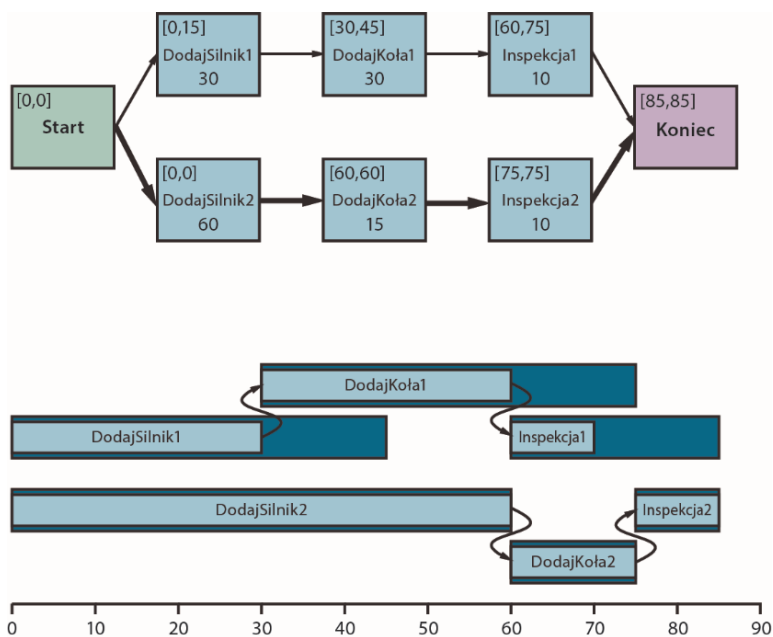
**RYSUNEK 11.4.** Schematy przykładowych zbiorów osiągalności. Zbiór stanów docelowych zaznaczony jest fioletowym tłem. Czarne i brązowe strzałki reprezentują możliwe implementacje wysokopoziomowych akcji (odpowiednio)  $h_1$  i  $h_2$ . (a) Zbiór osiągalności akcji  $h_1$  ze stanu  $s$  (b) Zbiór osiągalności ciągu akcji  $[h_1, h_2]$ ; ponieważ zbiór ten ma niepustą część wspólną ze zbiorem stanów docelowych, wspomniany ciąg prowadzi do osiągnięcia stanu docelowego



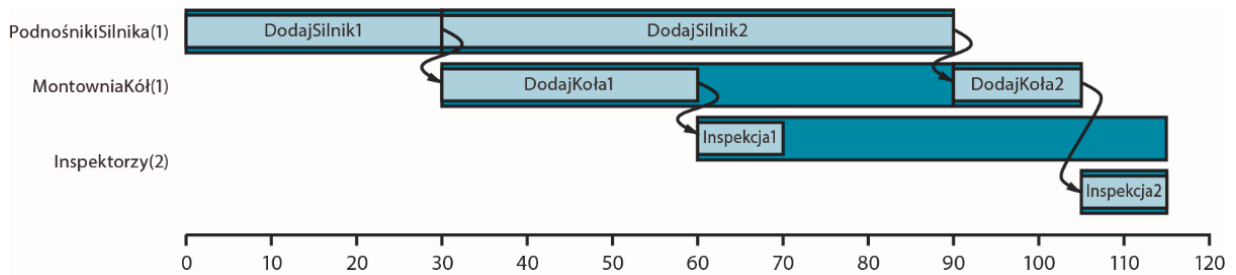
**RYSUNEK 11.5.** Osiągalność celów dla wysokopoziomowych planów z aproksymowanymi opisami. Zbiór stanów docelowych wyróżniony jest fioletowym tłem. Dla każdego planu pokazano zbiory osiągalności pesymistyczne (jasnoniebieskie tło, ciągła obwódka) i optymistyczne (szare tło, przerywana obwódka). (a) Plan reprezentowany przez czarną strzałkę definitywnie osiąga cel, podczas gdy plan reprezentowany przez strzałkę brązową definitywnie go nie osiąga. (b) Plan, który *być może* osiąga cel (optymistyczny zbiór osiągalności przecina się ze zbiorem stanów docelowych), ale *nie ma pewności*, że go osiąga (pesymistyczny zbiór osiągalności nie przecina się ze zbiorem stanów docelowych). Plan wymaga dalszego precyzowania w celu określenia, czy faktycznie osiąga cel, czy też definitywnie go nie osiąga.



**RYSUNEK 11.6.** Początkowo zakłada się, że ciąg akcji „cały plan” doprowadzi agenta ze stanu  $S$  do stanu  $G$ . Agent wykonuje kilka kroków planu i gdy spodziewa się, że osiągnął stan  $E$ , obserwuje, że znalazł się w stanie  $O$ . Wykonuje więc minimalną akcję naprawczą i kontynuuje plan w kierunku stanu  $G$



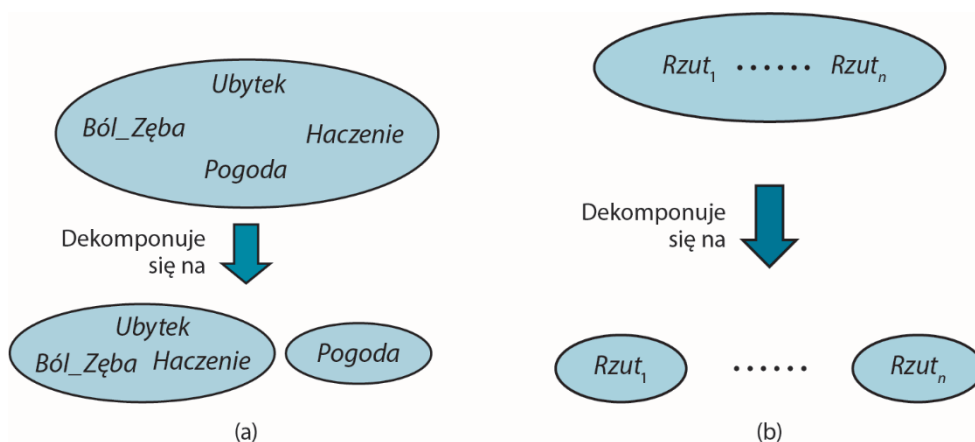
**RYSUNEK 11.7.** Część górna: reprezentacja ograniczeń czasowych problemu harmonogramowania z listingu 11.7. U dołu każdego prostokąta pokazano czas trwania danej akcji. W celu rozwiązania problemu obliczamy najwcześniejszy i najpóźniejszy moment rozpoczęcia danej akcji, czyli parę  $[ES, LS]$ , pokazaną w górnej części prostokątów. Różnica  $LS - ES$  jest marginesem danej akcji, akcje leżące na ścieżce krytycznej, wyróżnione pogrubionymi strzałkami, mają zerowy margines. Część dolna: to samo rozwiązanie odniesione do osi czasu. Niebieskie prostokąty wyznaczają okna czasowe, w których dana akcja musi zostać wykonana, jeśli mają być respektowane ograniczenia nałożone na kolejność akcji. Rozmiar wolnego obszaru niebieskiego prostokąta równy jest marginesowi danej akcji



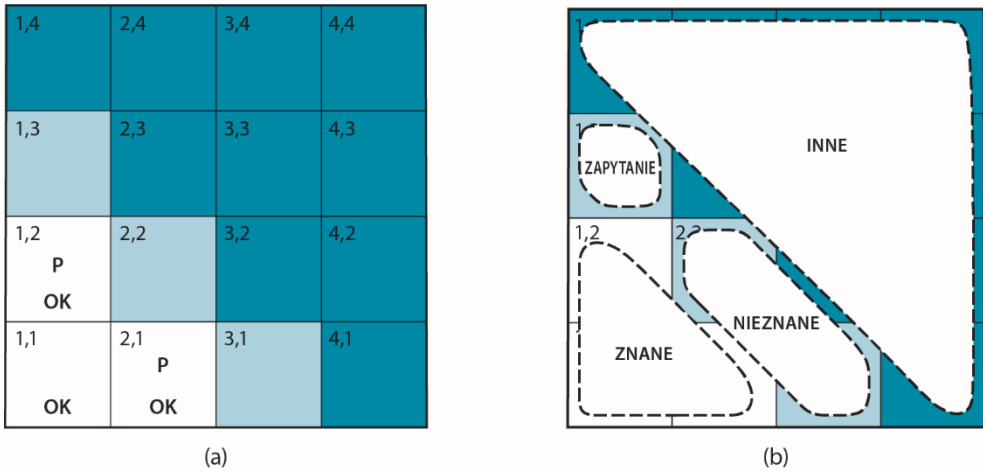
**RYSUNEK 11.8.** Rozwiązanie problemu planowania produkcji z listingu 11.7, uwzględniające ograniczenia nałożone na wykorzystywanie zasobów. W lewej części wymienione są trzy szeregowo wieloużywalne zasoby, w części prawej pokazane są akcje, z poziomym wyrównaniem z wykorzystywanymi zasobami. W świetle ograniczeń nałożonych na kolejność akcji i wykorzystywanie zasobów możliwe są dwa harmonogramy zależne od tego, która akcja jako pierwsza skorzysta z podnośnika silnika. Pokazaliśmy krótszy harmonogram, zakładający realizację zadania w ciągu 115 minut

## ROZDZIAŁ 12

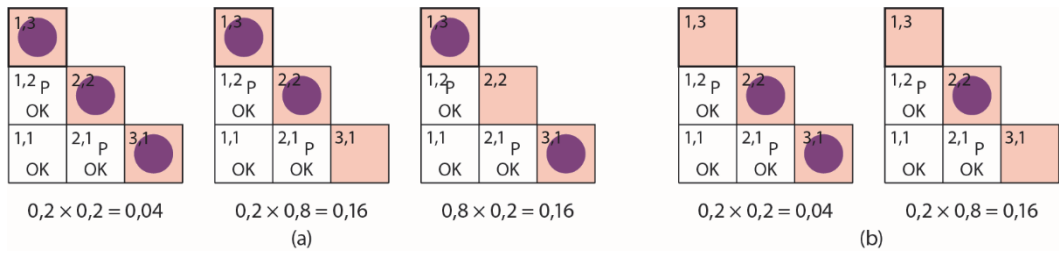
# KWANTYFIKOWANIE NIEPEWNOŚCI



**RYSUNEK 12.1.** Dwa przykłady faktoryzacji dużego rozkładu wspólnego na podstawie bezwzględnej niezależności. (a) Meteorologia i stomatologia są niezależne (b) Kolejne rzuty monetą są niezależne od siebie

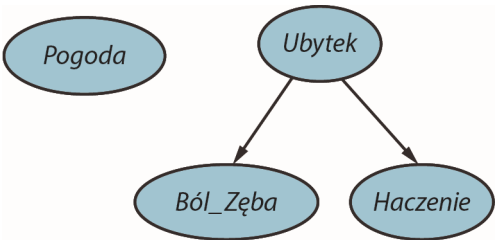


**RYSUNEK 12.2.** (a) Po stwierdzeniu przeciągu w kwadratach (1, 2) i (2, 1) agent nie ma bezpiecznego ruchu. (b) Podział zbioru kwadratów na znane, granicę i pozostałe, z perspektywy badania kwadratu (1, 3)

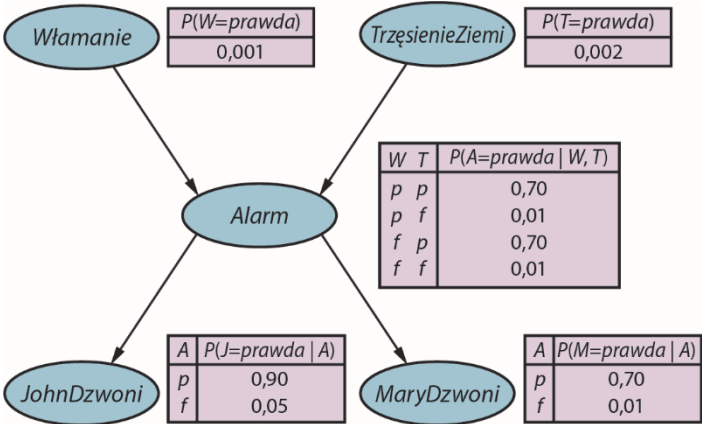


**RYSUNEK 12.3.** Spójne modele dla zmiennych granicy  $J_{2,2}$  i  $J_{3,1}$ , dla każdego modelu uwidoczniono  $P(granica)$ : (a) trzy modele dla  $J_{1,3} = \text{prawda}$  z dwiema lub trzema jamami, (b) dwa modele dla  $J_{1,3} = \text{fałsz}$  z jedną lub dwiema jamami

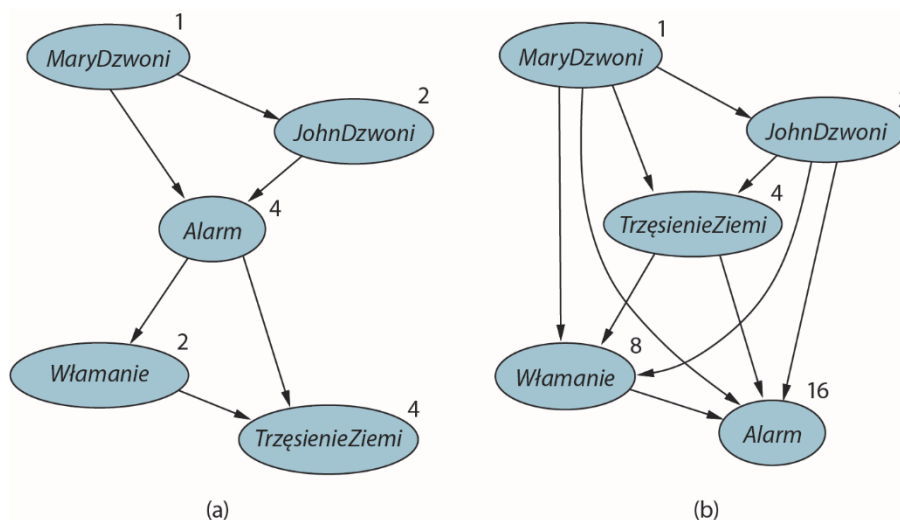
# WNIOSKOWANIE PROBABILISTYCZNE



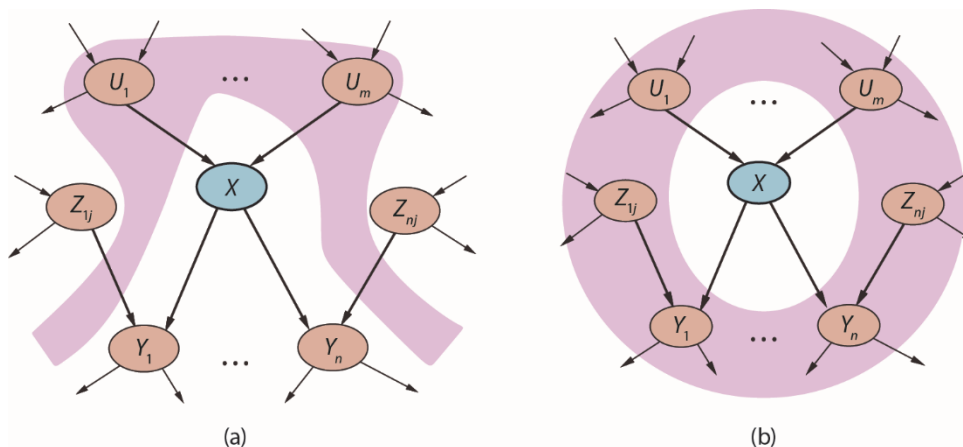
**RYSUNEK 13.1.** Prosta sieć bayesowska, której zmienna *Pogoda* jest niezależna od pozostałych trzech zmiennych, a zmienne *Ból\_Zęba* i *Haczenie* są warunkowo niezależne wobec bieżącej wartości zmiennej *Ubytek*



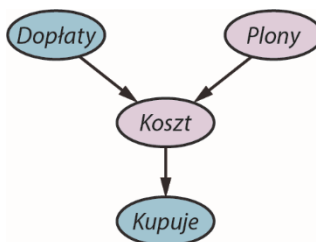
**RYSUNEK 13.2.** Typowa sieć bayesowska ukazująca zarówno topologię, jak i tablice prawdopodobieństw warunkowych (CPT). Litery *W*, *T*, *A*, *J* i *M* to skróty nazw zmiennych *Włamanie*, *TrzęsienieZiemi*, *Alarm*, *JohnDzwoni* i *MaryDzwoni*



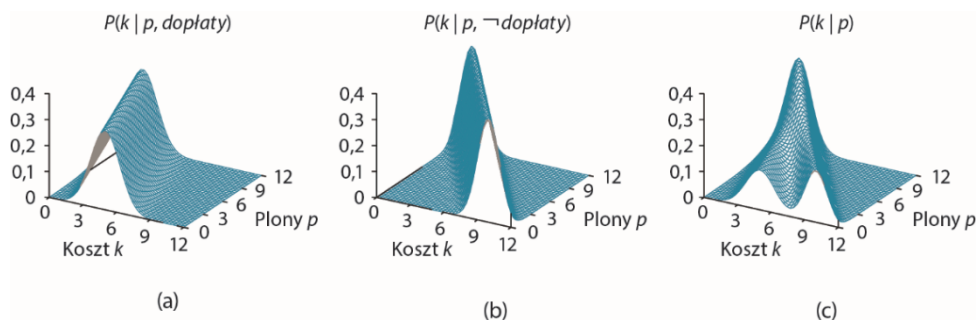
**RYСУNEK 13.3.** Struktura sieci i liczba parametrów w zależności od kolejności dodawania zmiennych. (a) Struktura otrzymana przy kolejności  $M, J, A, W, T$ . (b) Struktura otrzymana przy kolejności  $M, J, T, W, A$ . Dla każdego węzła podano liczbę wymaganych parametrów — łącznie 13 w konfiguracji (a) i 31 w konfiguracji (b). W konfiguracji z rysunku 13.2 wymagane było jedynie 10 parametrów



**RYСУNEK 13.4.** (a) Węzeł  $X$  jest warunkowo niezależny od węzłów  $Z_{ij}$  niebędących jego potomkami, wobec swych rodziców  $U_i$  (w różowych elipsach). (b) Węzeł  $X$  jest warunkowo niezależny od wszystkich innych węzłów sieci, wobec swego pokrycia Markowa (różowe elipsy)

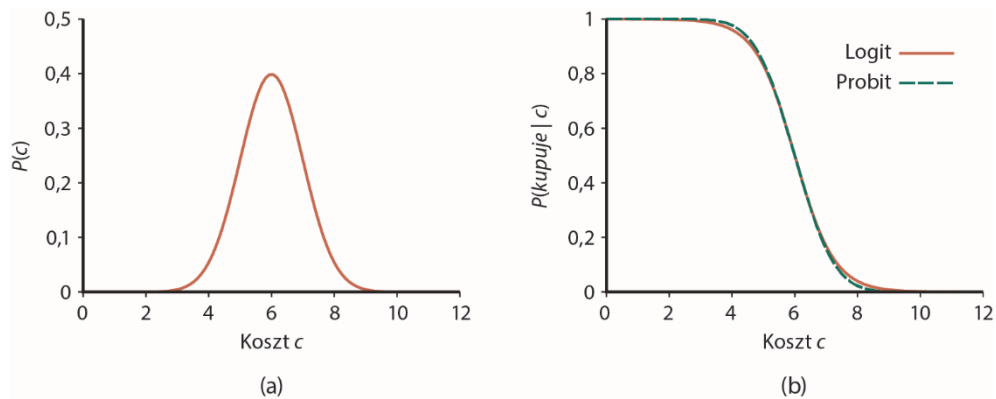


RYSUNEK 13.5. Prosta sieć ze zmiennymi dyskretnymi (*Dopłaty* i *Kupuje*) i ciągłymi (*Plony* i *Koszt*)

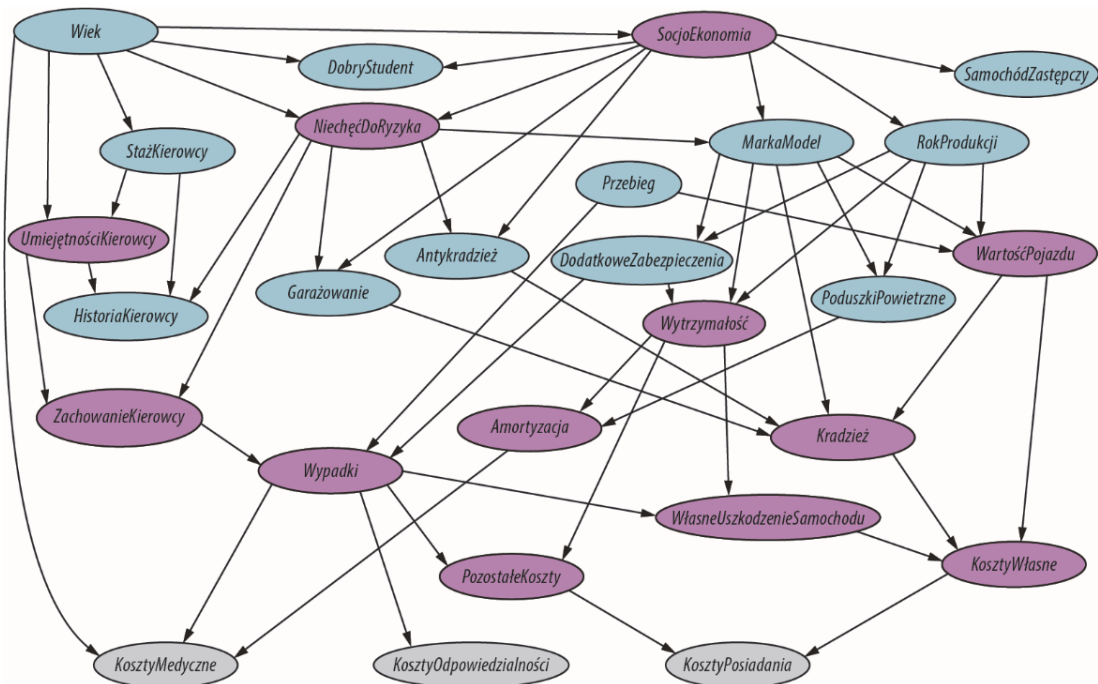


RYSUNEK 13.6. Wykresy (a) i (b) ukazują rozkład prawdopodobieństwa *Kosztu* w funkcji wielkości *Plonów*, wobec wartości *prawda* i *falsz* zmiennej *Dopłaty*. Na wykresie (c) widoczny jest rozkład  $P(\text{Koszt} | \text{Plony})$  otrzymany przez zsumowanie wartości z wykresów (a) i (b)

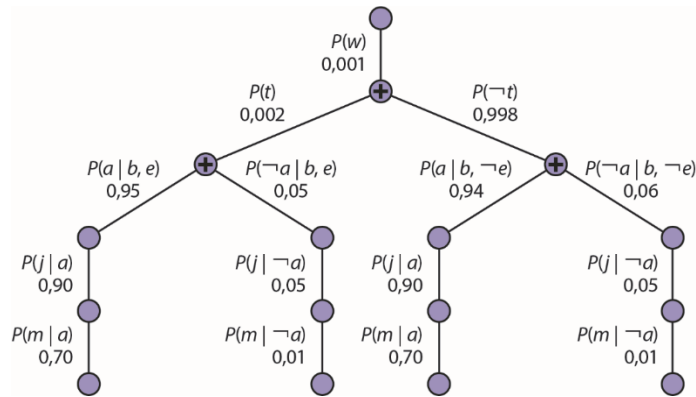




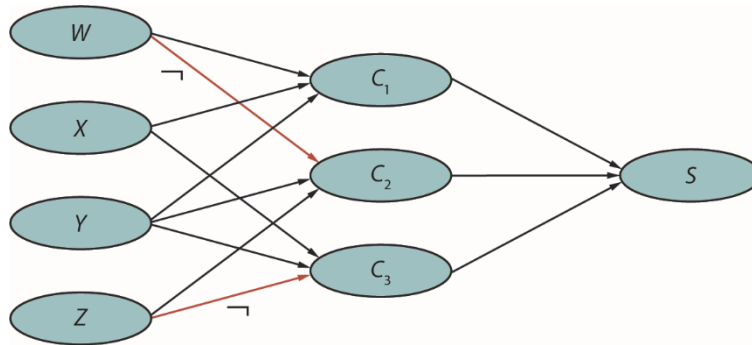
RYSUNEK 13.7. (a) Rozkład normalny (Gaussa) dla progu kosztu, wyśrodkowany dla  $\mu = 6,0$  i odchylenia standardowego  $\sigma = 1,0$ . (b) Modele expitowy i probitowy dla prawdopodobieństwa *kupuje* wobec *koszt*, dla parametrów  $\mu = 6,0$  i  $\sigma = 1,0$



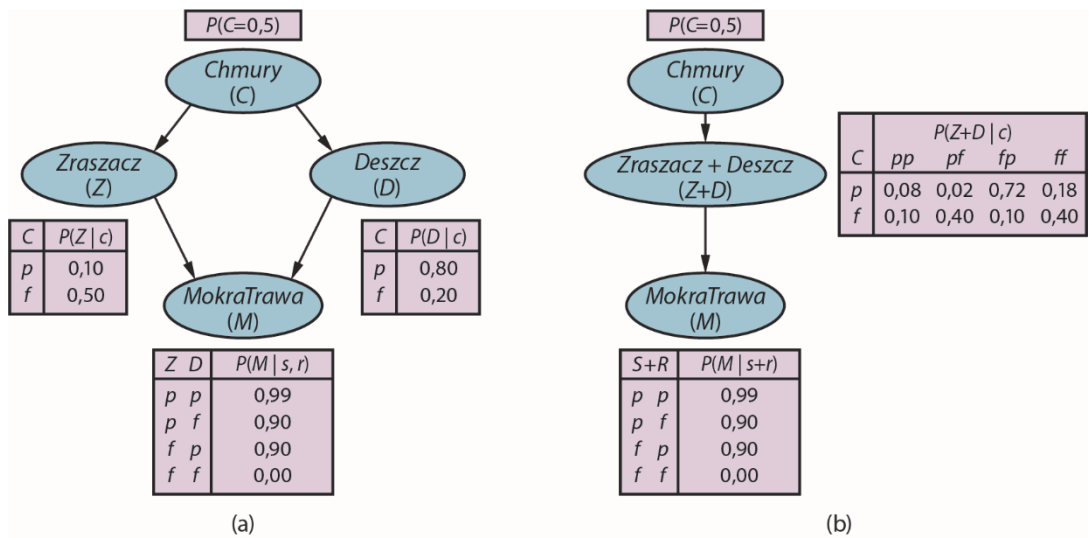
RYSUNEK 13.8. Sieć bayesowska na potrzeby ewaluacji aplikacji do ubezpieczania samochodów



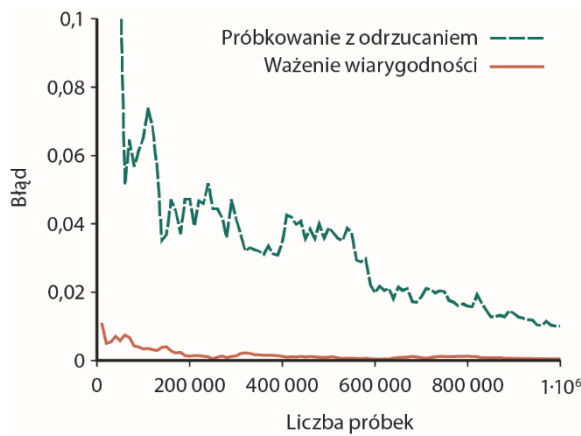
**RYСУNEK 13.9.** Struktura wyrażenia w równaniu (13.5). Obliczanie wartości przebiega w kierunku zstępującym (od góry do dołu) przez mnożenie wartości na każdej ścieżce i sumowanie w węzłach  $\oplus$ . Zauważmy powtarzające się ścieżki dla  $j$  i  $m$



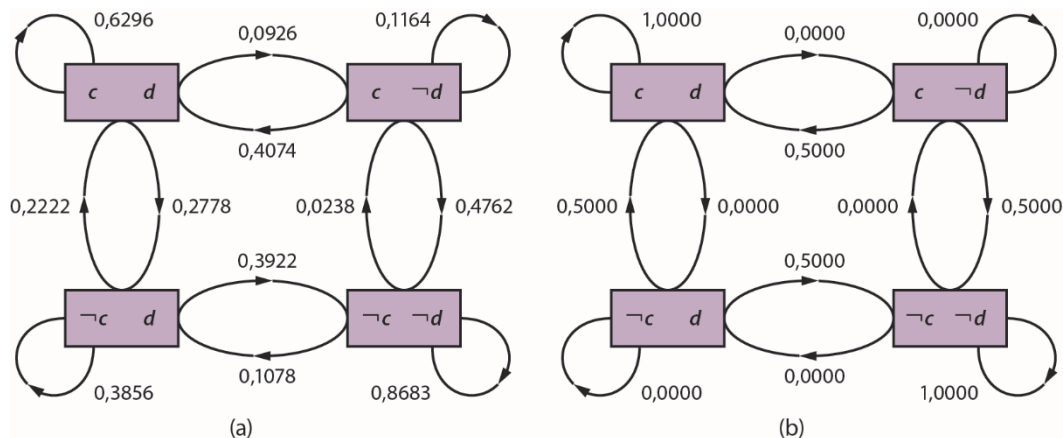
**RYСУNEK 13.10.** Sieć bayesowska kodująca zdanie 3-CNF  $(W \vee X \vee Y) \wedge (\neg W \vee Y \vee Z) \wedge (X \vee Y \vee \neg Z)$ .



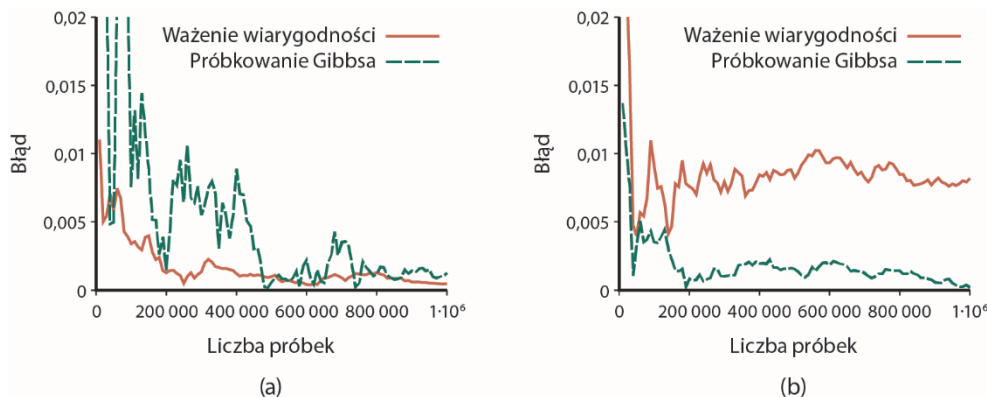
**RYСУNEK 13.11.** (a) Sieć wielokrotnie połączona, opisująca zwyczajowe zachowanie Mary w jej ogródku: każdego ranka ocenia ona prawdopodobieństwo nadejścia opadów i gdy niebo jest zachmurzone, zwykle nie włącza zraszacza. Z kolei włączony zraszacz lub padający deszcz powodują, że trawa staje się mokra. Zatem zmienna *Chmury* wywiera wpływ na zmienną *MokraTrawa* poprzez dwie różne ścieżki. (b) Klasteryzowany odpowiednik sieci wielokrotnie połączonej



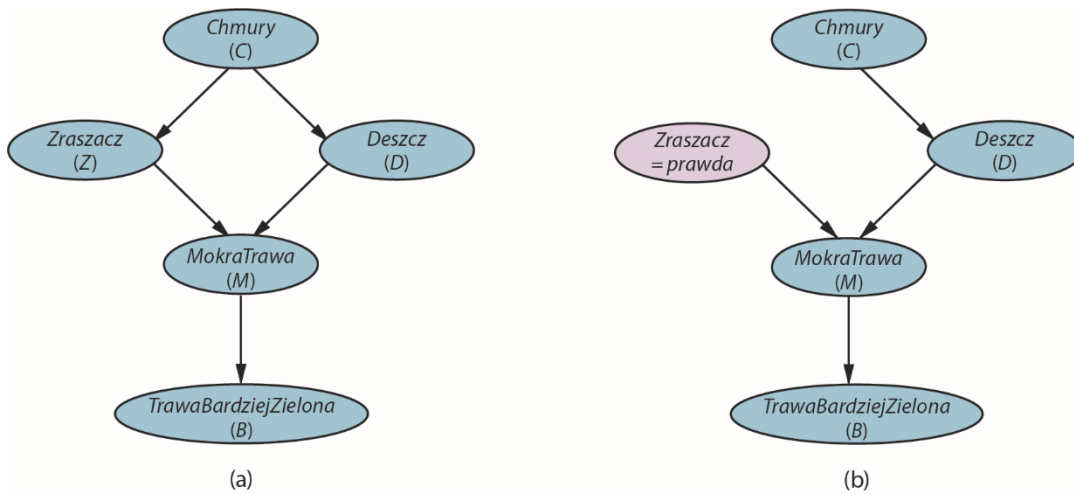
**RYСУNEK 13.12.** Wydajność próbkowania z odrzucaniem i ważenia wiarygodności w sieci z rysunku 13.8. Oś *x* reprezentuje liczbę generowanych próbek, oś *y* reprezentuje błąd bezwzględny obu aproksymacji dla zapytania *KosztyPosiadania*



**RYСУNEK 13.13.** (a) Stany i prawdopodobieństwa przejść między nimi w łańcuchu Markowa dla zapytania  $P(\text{Deszcz} \mid \text{Zrascacz} = \text{prawda}, \text{MokraTrawa} = \text{prawda})$ . Zwróćmy uwagę na „samozapętnienia”: stan pozostaje niezmieniony, niezależnie od wyboru konkretnej zmiennej, po czym ponowne próbkowanie tej zmiennej zwraca jej dotychczasową wartość. (b) Prawdopodobieństwa przejść w przypadku, gdy tablica CPT dla zmiennej  $\text{Deszcz}$  wymusza przyjęcie przez nią wartości identycznej z wartością zmiennej  $\text{Chmury}$



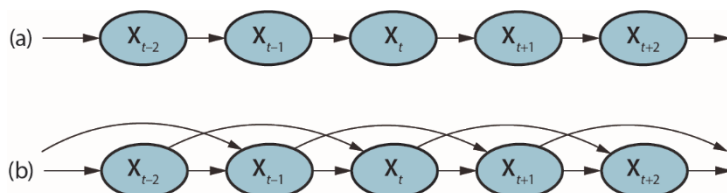
**RYСУNEK 13.14.** Porównanie wydajności próbkowania Gibbsa i ważenia wiarygodności dla sieci z rysunku 13.8 (a) dla standardowego zapytania  $\text{KosztPosiadania}$ , (b) dla przypadku, gdy zmienne wyjściowe są obserwowane i zmienna  $\text{Wiek}$  jest zmienną zapytania.



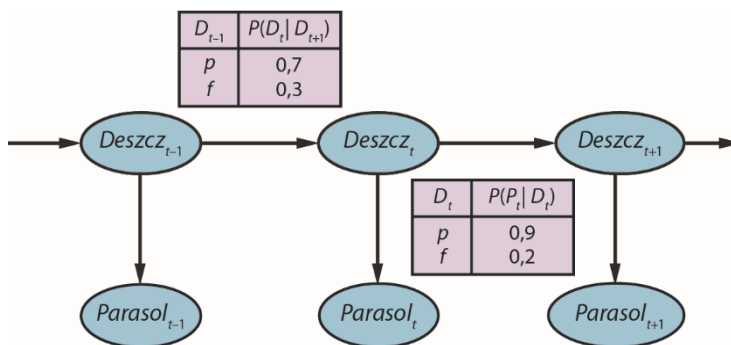
**RYСУNEK 13.15.** (a) Przyczynowa sieć bayesowska reprezentująca relacje „przyczyna-skutek” w zbiorze pięciu zmiennych. (b) Ta sama sieć po wykonaniu akcji włączenia *Zraszacza*

## ROZDZIAŁ 14

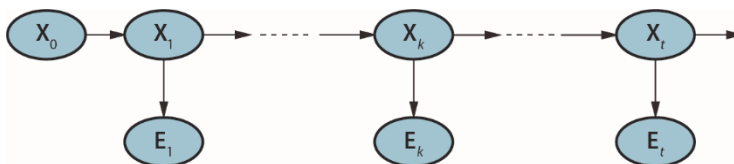
# PROBABILISTYCZNE WNIOSKOWANIE W CZASIE



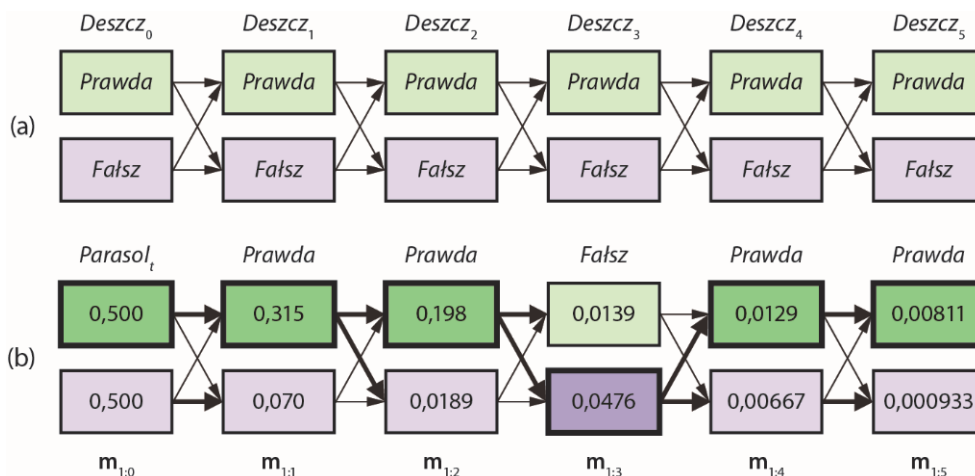
**RYSUNEK 14.1.** (a) Struktura sieci bayesowskiej odpowiadającej procesowi Markowa pierwszego rzędu ze stanem definiowanym przez zmienne  $X_t$ . (b) Proces Markowa drugiego rzędu



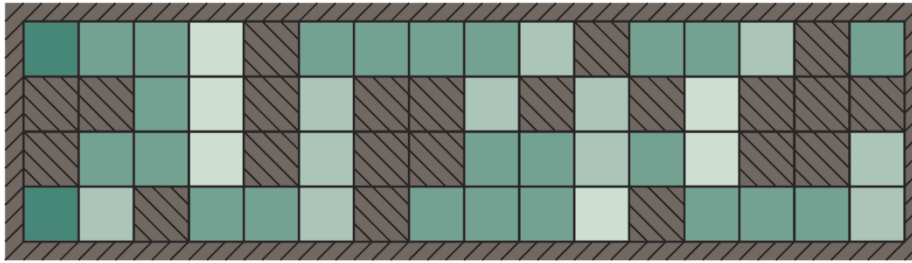
**RYSUNEK 14.2.** Struktura sieci bayesowskiej i rozkłady warunkowe opisujące problem z prognozowaniem opadów na podstawie otwartego parasola. Model przejścia  $P(Deszcz_t | Deszcz_{t-1})$  i model sensoryczny  $P(Parasol_t | Deszcz_t)$



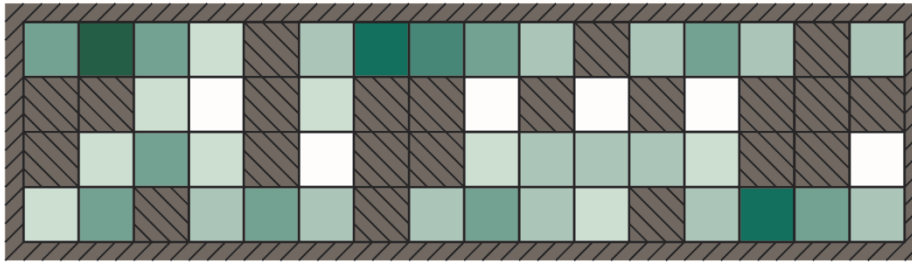
**RYСУNEK 14.3.** Wygładzanie obliczające  $P(X_k | e_{1:t})$  — rozkład a posteriori stanu w minionej chwili  $k$ , na podstawie ciągu obserwacji w chwilach od 1 do  $t$



**RYСУNEK 14.4.** (a) Możliwy ciąg stanów  $Deszcz_t$  może być postrzegany jako ścieżka w grafie możliwych stanów w poszczególnych krokach czasowych (stany reprezentowane są w formie prostokątów, by nie mylić ich z elipsami reprezentującymi węzły sieci). (b) Operacje wykonywane przez algorytm Viterbiego w problemie z parasolem, dla ciągu obserwacji [prawda, prawda, fałsz, prawda, prawda] rozpoczynającego się w chwili 1. Dla poszczególnych  $t$  pokazaliśmy wartości komunikatu  $m_{1:t}$ , stanowiące prawdopodobieństwa osiągnięcia stanu  $t$  poprzez najlepszy ciąg. Również dla każdego stanu wiodąca do niego pogrubiona strzałka rozpoczyna się w jego najlepszym poprzedniku — w sensie największego iloczynu prawdopodobieństwa poprzedzającego ciągu i prawdopodobieństwa modelu przejścia. Cofając się wzdłuż pogrubionych strzałek, począwszy od najbardziej prawdopodobnego stanu  $m_{1:5}$ , przemierzamy wstecz najbardziej prawdopodobny ciąg — jego elementy wyróżnione są pogrubionymi obwódkami i bardziej intensywnym tłem



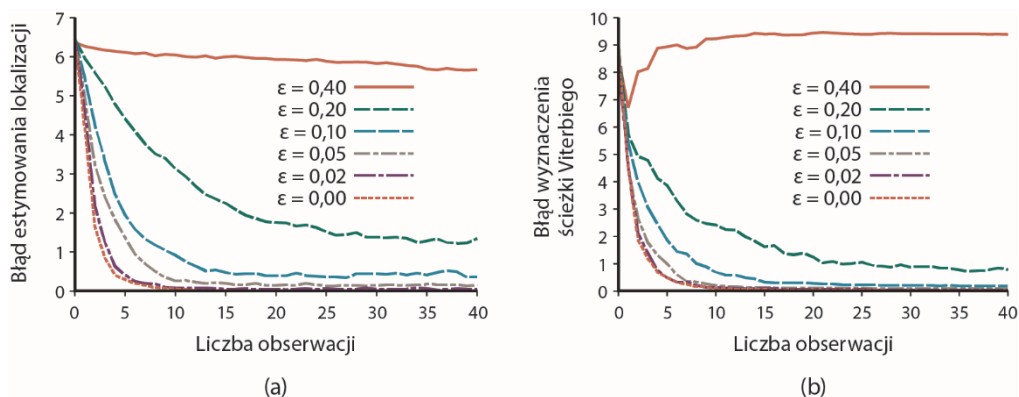
(a) Rozkład a posteriori prawdopodobieństw lokalizacji robota po zaobserwowaniu  $E_1 = 1011$



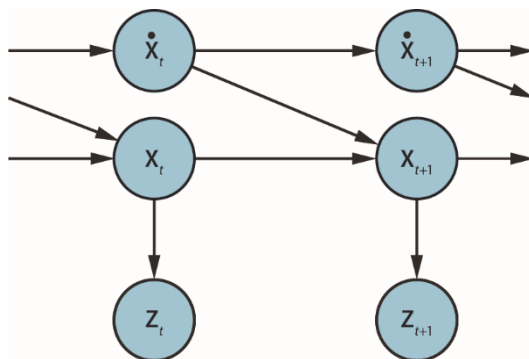
(b) Rozkład a posteriori prawdopodobieństw lokalizacji robota po zaobserwowaniu  $E_1 = 1011, E_2 = 1010$

**RYSUNEK 14.5.** Prawdopodobieństwo a posteriori lokalizacji robota. (a) Po jednej obserwacji  $E_1 = 1011$  (przeszkoda znajduje się od północy, południa i zachodu). (b) Po losowym przemieszczeniu się robota na przyległą pozycję i drugiej obserwacji  $E_2 = 1010$  (przeszkody od północy i południa). Stopień zaciemnienia danego kwadratu odpowiada prawdopodobieństwu, że jest on bieżącą lokalizacją robota. Prawdopodobieństwo zawodności przekłamania czujnika wynosi  $\varepsilon = 0,2$  niezależnie dla każdego bitu

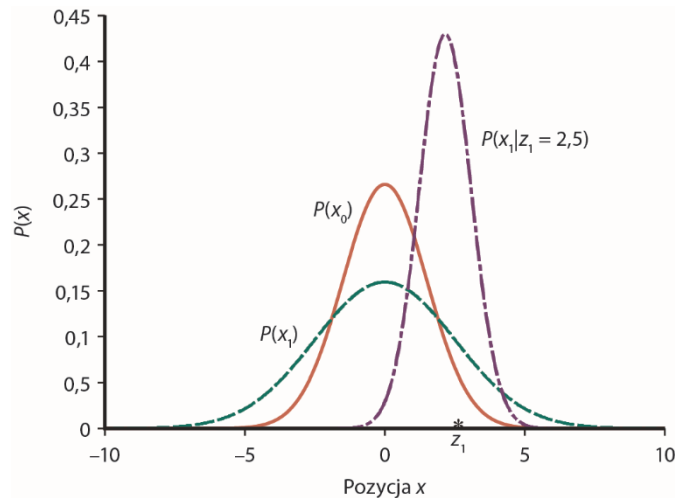




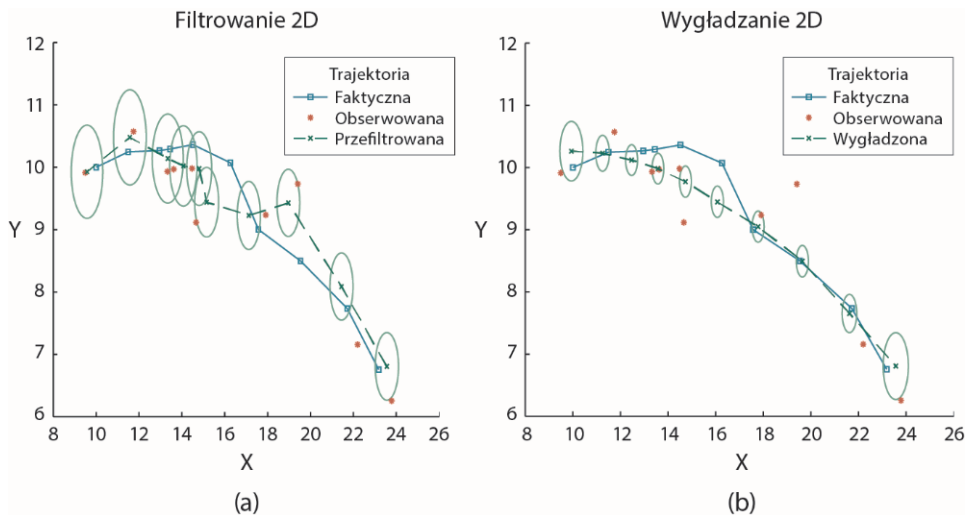
**RYSUNEK 14.6.** Dokładność modelu HMM w funkcji długości ciągu obserwacji i różnych współczynników zawadności czujnika  $\varepsilon$ ; dane zostały uśrednione po 400 przebiegach. (a) Błąd estymacji lokalizacji definiowany według metryki odległości Manhattanu. (b) Błąd wyznaczenia ścieżki Viterbiego, jako uśredniona odległość między stanami estymowanymi a położonymi na rzeczywistej ścieżce, definiowany według metryki odległości Manhattanu



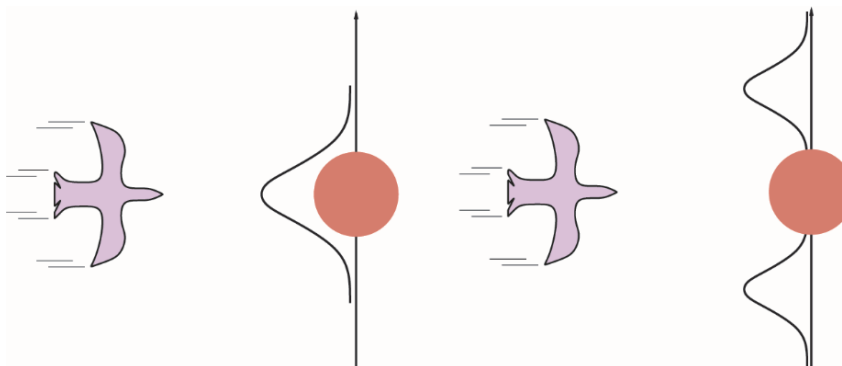
**RYSUNEK 14.7.** Struktura sieci bayesowskiej dla liniowego systemu dynamicznego definiowanego przez zmienne pozycji  $X_t$ , prędkości  $\dot{X}_t$  i obserwacji  $Z_t$



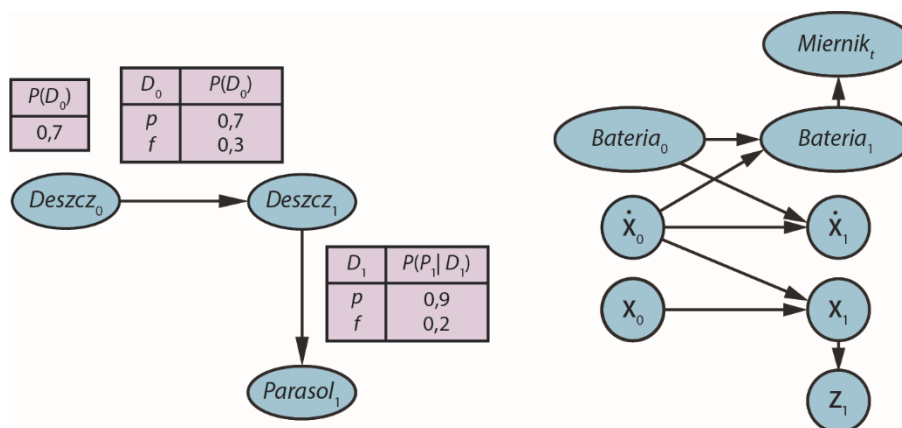
**RYSUNEK 14.8.** Etapy cyklu aktualizacji w filtracji Kalmana dla spaceru losowego z prawdopodobieństwem a priori ( $\mu_0 = 0,0$ ,  $\sigma_0 = 1,5$ ), szumem przejścia  $\sigma_x = 2,0$ , szumem czujnika  $\sigma_z = 1,0$  i pierwszą obserwacją  $z_1 = 2,5$  (zaznaczoną na osi  $x$ ). Zwróćmy uwagę na spłaszczenie przewidywania  $P(x_1)$  względem  $P(x_0)$ , wywołane przez szum przejścia. Zauważmy również, że średnia rozkładu a posteriori  $P(x_1 | z_1)$  przesunięta jest nieznacznie w lewo względem obserwacji  $z_1$ , ponieważ średnia ta stanowi wynik ważonego uśrednienia przewidywania i obserwacji



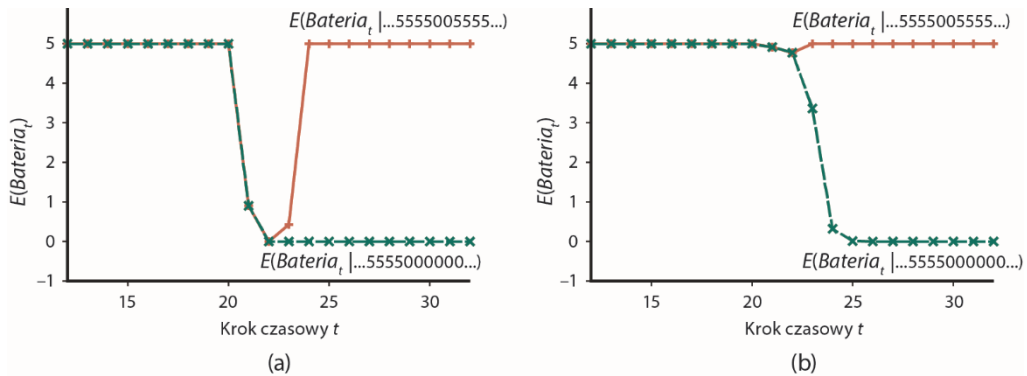
**RYSUNEK 14.9.** (a) Rezultaty filtracji Kalmana dla obiektu przemieszczającego się w płaszczyźnie X-Y: rzeczywista trajektoria (od lewej do prawej), seria zakłóconych obserwacji i trajektoria estymowana przez filtrację Kalmana. Elipsy reprezentują wariancję estymacji położenia. (b) Rezultaty wygładzania Kalmana dla tego samego ciągu obserwacji



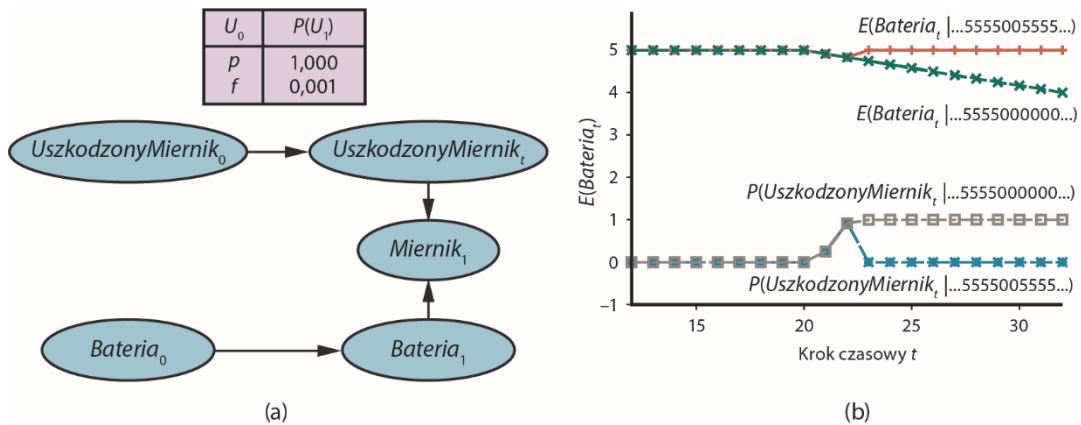
**RYSUNEK 14.10.** Ptak zmierzający wprost w kierunku przeszkody (widok z góry). (a) Przewidywanie filtrowania Kalmana wykorzystujące pojedynczy rozkład Gaussa — ptak uderzy w przeszkodę. (b) Bardziej realistyczny model, uwzględniający wykonanie przez ptaka nagłego zwrotu w celu ominięcia przeszkody



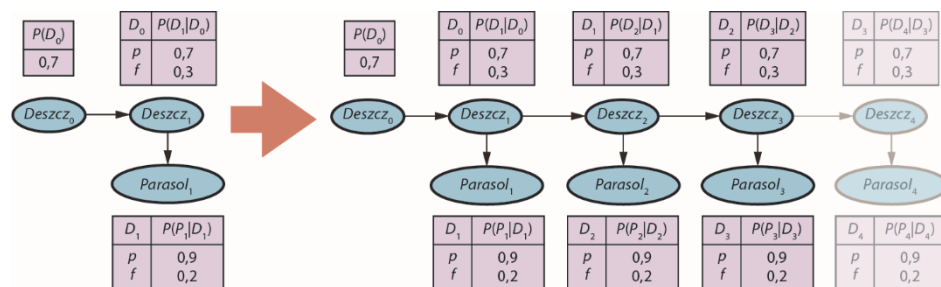
**RYSUNEK 14.11.** Z lewej: specyfikacja prawdopodobieństwa a priori, modelu przejścia i modelu sensorycznego dla sieci DBN odzwierciedlającej problem z parasolem. Kolejne kroki czasowe są kopiami kroku 1. Z prawej: prosta sieć DBN dla robota poruszającego się w płaszczyźnie  $X$ - $Y$



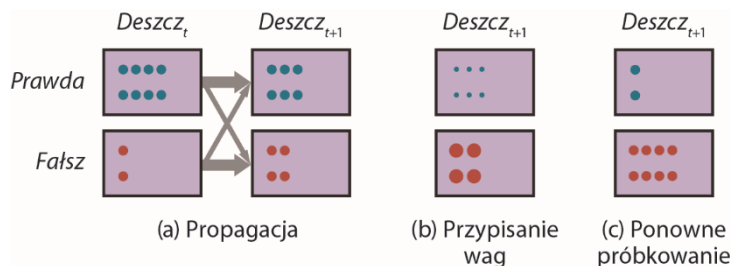
**RYSUNEK 14.12.** (a) Górna krzywa: trajektoria oczekiwanych wartości  $Bateria_t$  dla ciągu obserwacji złożonego z samych 5, z wyjątkiem kroków  $t = 21$  i  $t = 22$ ; wartości estymowane przy użyciu prostego gaussowskiego modelu błędów. Dolna krzywa: trajektoria dla przypadku, gdy w kroku  $t = 21$  obserwacje zmieniają się na 0 i takimi pozostają. (b) Ten sam eksperyment z punktu widzenia modelu przejściowych usterek. Przejściowa usterka miernika jest obsługiwana poprawnie, natomiast jego permanentna awaria wywołuje nadmierny pesymizm w kwestii stanu naładowania baterii



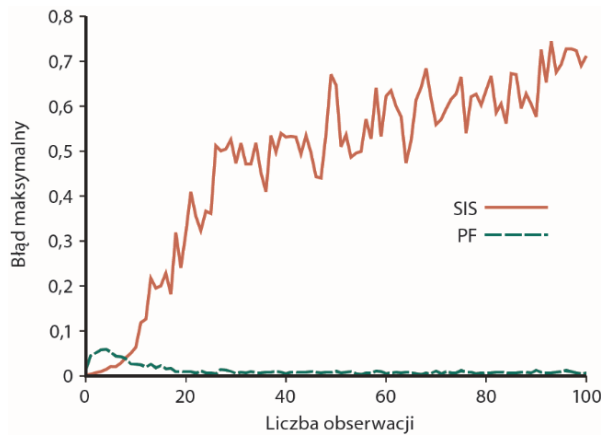
**RYSUNEK 14.13.** (a) Fragment sieci DBN ukazujący wartość zmiennej reprezentującej status miernika, niezbędnej dla modelowania trwałych awarii tegoż miernika. (b) Górne krzywe: trajektorie oczekiwanej wartości zmiennej  $Bateria_t$ , dla obserwacji zwiastujących przejściową usterkę i trwałą awarię miernika. Dolne krzywe: trajektorie prawdopodobieństw dla zmiennej  $UszkodzonyMiernik_t$  w świetle wspomnianych ciągów obserwacji



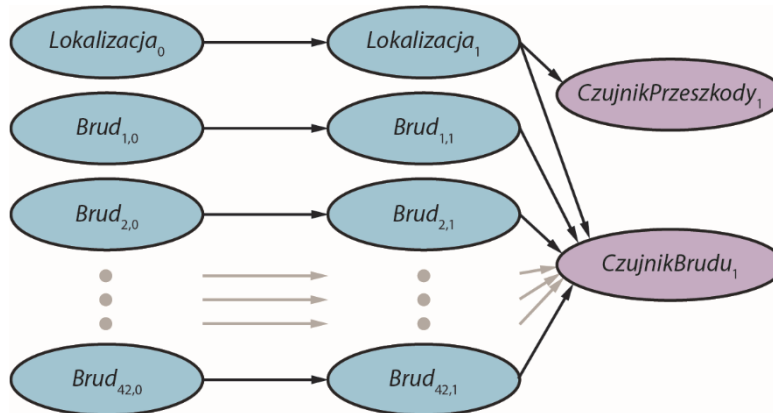
**RYSUNEK 14.14.** Rozwijanie dynamicznej sieci bayesowskiej — dane pierwszego kroku czasowego są powielane w celu przystosowania dociągu obserwacji  $Parasol_{1:3}$ . Dodawanie kroku  $t = 4$  i następnych nie daje żadnego efektu w świetle dostępnych obserwacji



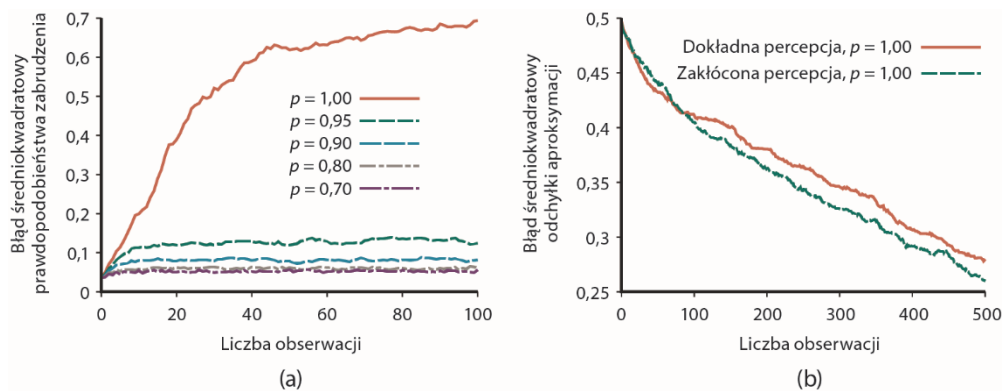
**RYSUNEK 14.15.** Cykl aktualizacji w filtrowaniu cząstek dla sieci DBN odzwierciedlającej problem z parasolem, dla liczby cząstek  $N = 10$ , z ukazaniem populacji próbek dla każdego stanu. (a) W chwili  $t$  8 próbek wskazuje na deszcz, a 2 na jego brak. Każda próbka jest propagowana w przód, przez próbkowanie nowego stanu poprzez model przejścia. W czasie  $t+1$  sześć próbek wskazuje na deszcz, a cztery na jego brak. (b) W czasie  $t+1$  zaobserwowano  $Parasol=falsz$ . Każda próbka opatrywana jest wagą proporcjonalną do wiarygodności obserwacji, co odzwierciedlone jest przez zróżnicowanie wielkości kółek. (c) Generowany jest nowy zbiór 10 próbek przez ważony wybór z bieżącego zbioru, w wyniku czego otrzymujemy 2 próbki wskazujące na deszcz i 8 na jego brak



**RYSUNEK 14.16.** Błąd maksymalny estymacji położenia robota w labiryncie (w porównaniu ze ścisłym wnioskowaniem) dla ważenia wiarygodności (sekwencyjnego próbkowania istotnościowego) z  $N = 100\,000$  próbek oraz filtrowania cząstek z  $N = 1000$  próbek. Dane są uśrednieniem z 50 przebiegów



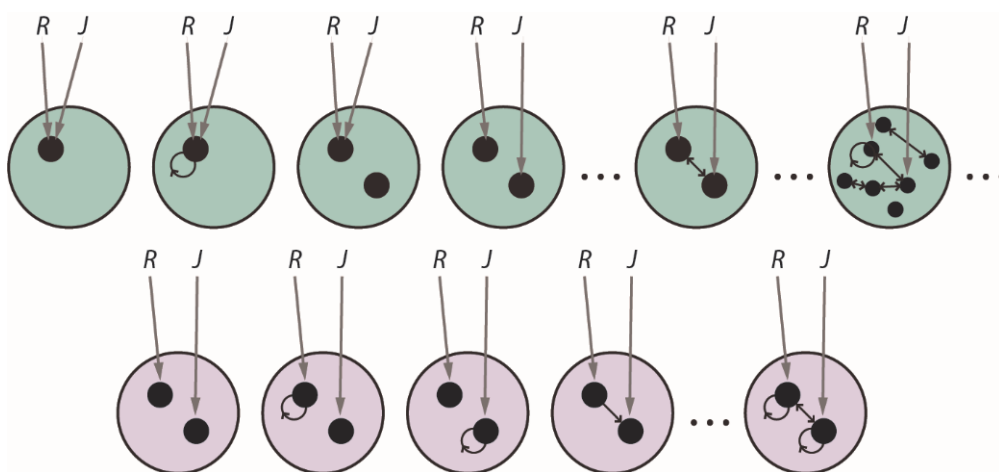
**RYSUNEK 14.17.** Dynamiczna sieć bayesowska dla lokalizacji z jednoczesnym mapowaniem w labiryncie ze stochastycznymi zabrudzeniami. Zabrudzenie utrzymuje się w kwadracie z prawdopodobieństwem  $p$ , a czysty kwadrat ulega zabrudzeniu z prawdopodobieństwem  $1 - p$ . Robot dysponuje czujnikiem zabrudzenia kwadratu, w którym aktualnie się znajduje; niezawodność tego czujnika wynosi 90%



**RYSUNEK 14.18.** (a) Wyniki standardowej wersji algorytmu filtrowania cząstek z 1000 cząstek. Widoczny jest błąd średniokwadratowy marginesowego prawdopodobieństwa zabrudzenia w porównaniu ze ścisłym wnioskowaniem, dla różnych wartości  $p$  utrzymywania się zabrudzenia. (b) Wyniki filtrowania cząstek Rao-Blackwellizacją (100 cząstek) w porównaniu z rzeczywistymi wartościami, w wersji dla niezawodnego czujnika lokalizacji i zakłócanego czujnika przeszkód, w warunkach deterministycznych zabrudzeń. Dane uśrednione po 20 przebiegach

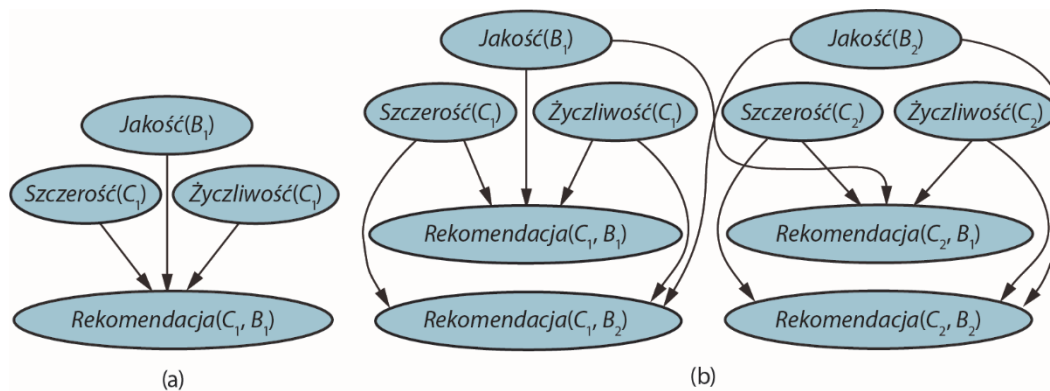
## ROZDZIAŁ 15

# PROGRAMOWANIE PROBABILISTYCZNE

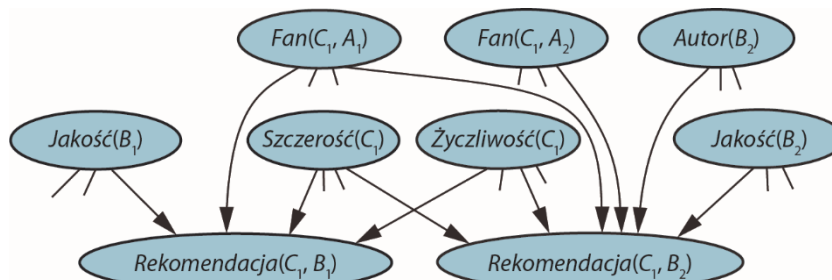


**RYSUNEK 15.1.** Górna część: niektóre z możliwych światów dla języka z dwoma symbolami stałymi —  $R$  i  $J$  — i jednego symbolu relacji binarnej, według standardowej semantyki logiki pierwszego rzędu. Dolna część: możliwe światy według semantyki bazodanowej. Interpretacja symboli stałych jest ustalona, każdy symbol stały odpowiada innemu obiektowi.

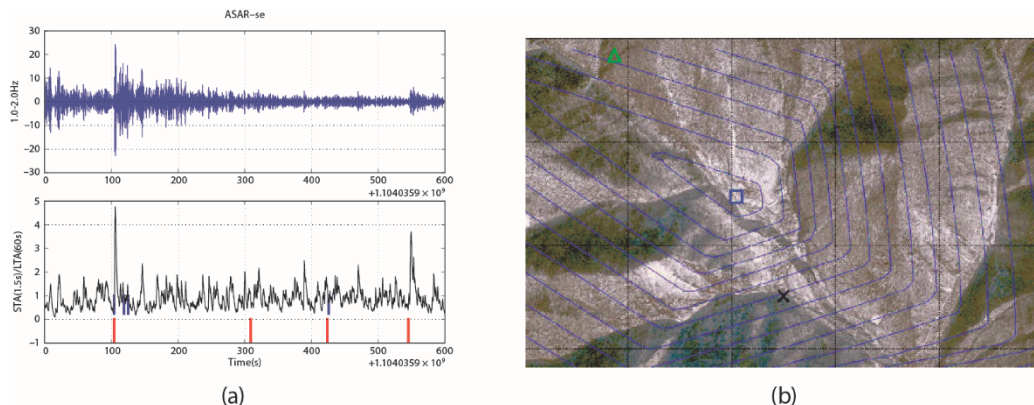




**RYСУNEK 15.2.** (a) Sieć bayesowska dla pojedynczego klienta  $C_1$  rekomendującego jedną książkę  $B_1$ . Zmienna  $Szczerość(C_1)$  jest zmienną boolowską, pozostałe zmienne przyjmują wartości całkowite od 1 do 5. (b) Sieć bayesowska dla dwóch klientów i dwóch książek

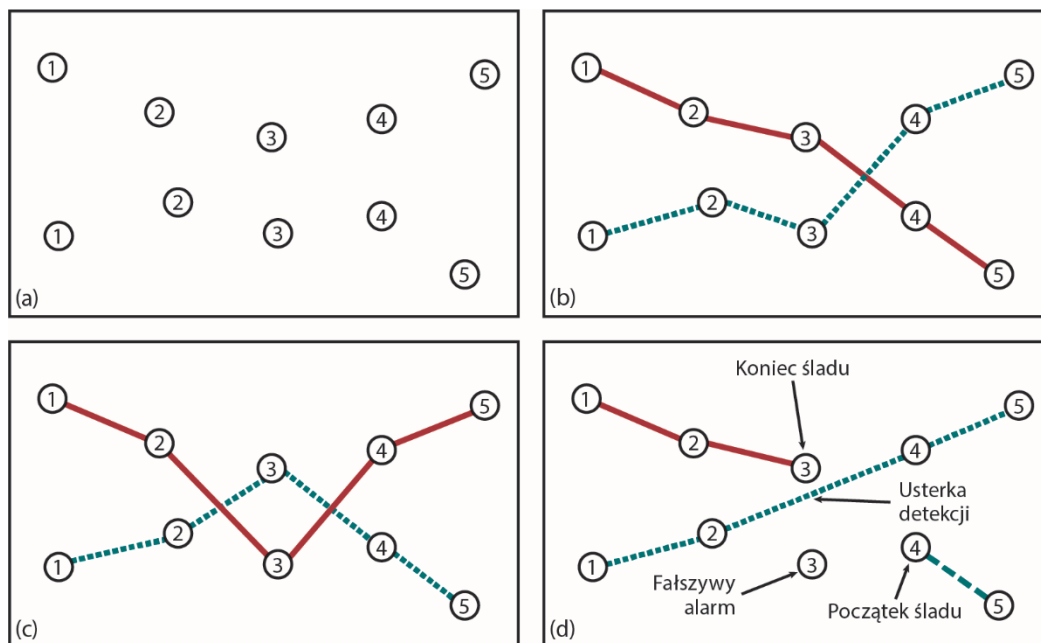


**RYСУNEK 15.3.** Fragment sieci bayesowskiej równoważnej przypadkowi, gdy autor książki  $B_2$  jest nieznan



**RYSUNEK 15.4.** (a) Górna część: przebieg falowy sygnału zarejestrowanego w wyniku trzęsienia ziemi w Alice Springs w Australii. Część dolna: sygnał po przetworzeniu mającym na celu zidentyfikowanie czasu przybycia fali sejsmicznej do czujnika. (b) Oszacowanie lokalizacji testu jądrowego przeprowadzonego w Korei Północnej 12 lutego 2013 roku. Zielony trójkąt w pobliżu lewego górnego narożnika fotografii to oznaczenie lokalizacji w biuletynie *UN CTBTO Late Event*, środek niebieskiego kwadratu to lokalizacja centrum eksplozji estymowana przez NET-VISA. Czarnym krzyżykiem oznaczono wejście do podziemnego centrum badawczego, w którym zlokalizowane są czujniki (ok. 75 kilometrów od estymowanego centrum eksplozji). Niebieskie kontury symbolizują rozkład prawdopodobieństwa a posteriori lokalizacji centrum eksplozji, estymowany przez NET-VISA

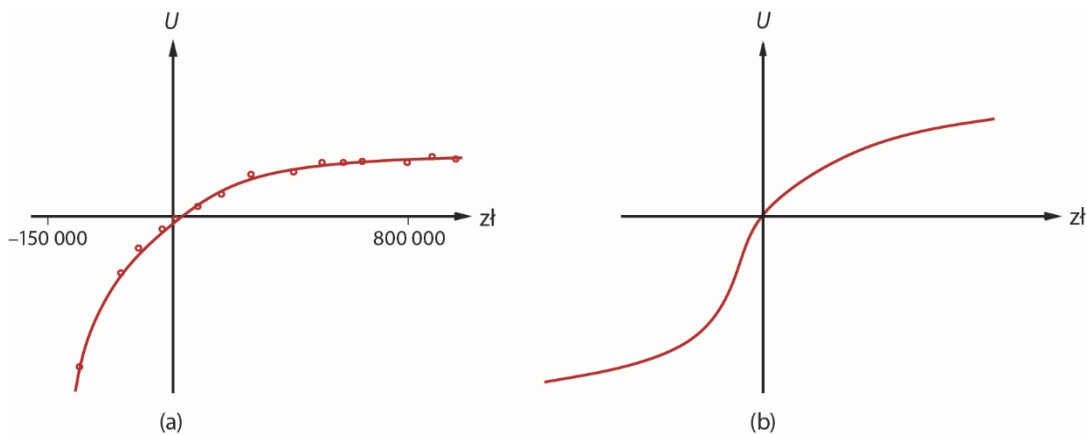
Dzięki uprzejmości *CTBTO Preparatory Commission*



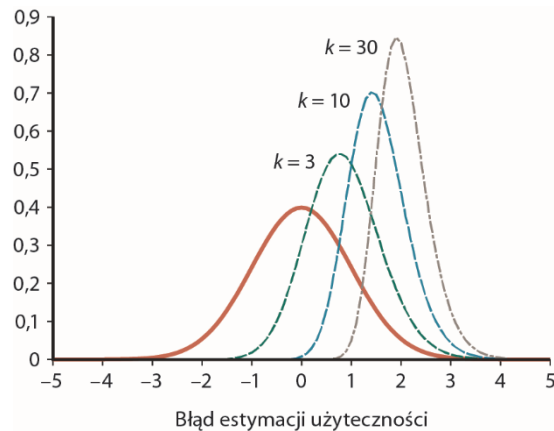
**RYСУNEK 15.5.** (a) Obserwacje położenia obiektów w przestrzeni 2D w pięciu krokach czasowych. Obserwacje (rozbityski) etykietowane są numerami kroków czasowych, lecz nie ma informacji umożliwiającej jednoznaczne przyporządkowanie poszczególnych obserwacji do konkretnych obiektów. (b) i (c) Dwa hipotetyczne skojarzenia śladów z obiektami. (d) Hipotetyczne skojarzenia śladów z obiektami w sytuacji, gdy możliwe jest wystąpienie fałszywego alarmu, usterki detekcji i przerwania obserwacji.

## ROZDZIAŁ 16

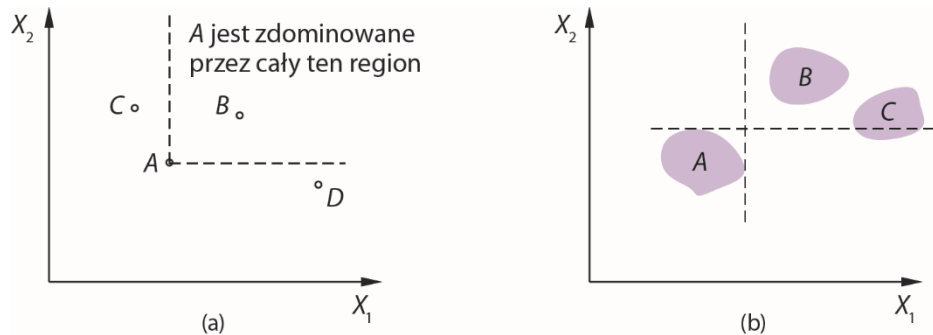
# PODEJMOWANIE PROSTYCH DECYZJI



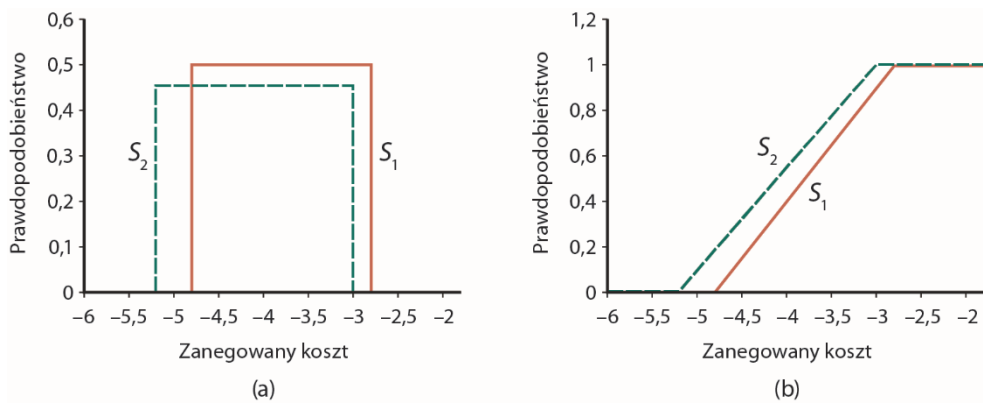
**RYSUNEK 16.2.** Użyteczność pieniędzy. (a) Wykres na podstawie przykładowych danych empirycznych w ograniczonym przedziale wartości. (b) Typowy wykres dla wszystkich wartości.



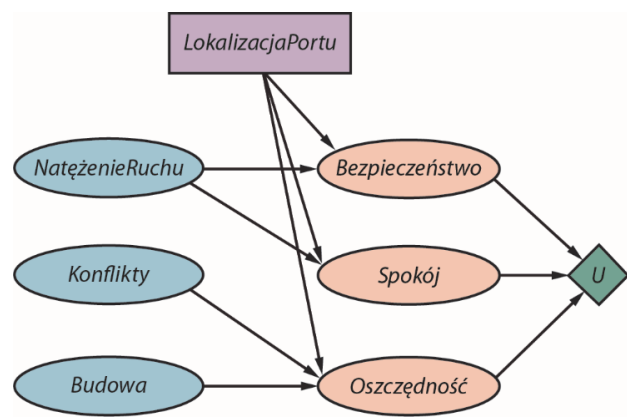
**RYSUNEK 16.3.** Nieuzasadniony optymizm wywołany wyborem najlepszej spośród  $k$  opcji. Zakładamy, że rzeczywista użyteczność każdej akcji równa jest 0, a jej estymacja podlega znormalizowanemu rozkładowi normalnemu (wyróżniona krzywa). Pozostałe krzywe ilustrują estymowane rozkłady dla  $k = 3, 10$  i  $30$ .



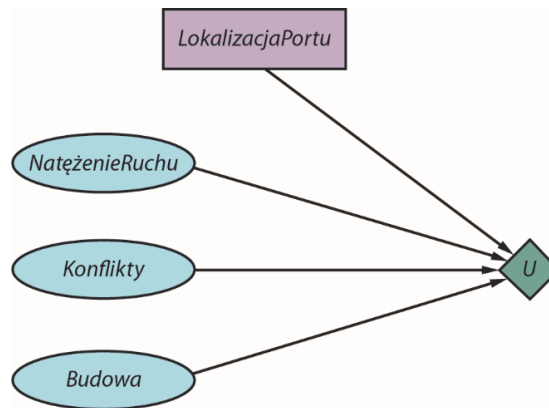
**RYSUNEK 16.4.** Ścisła dominacja. (a) Przypadek deterministyczny: opcja  $A$  jest ściśle zdominowana przez  $B$ , lecz nie przez  $C$  ani  $D$ . (b) Przypadek niepewności: opcja  $A$  jest ściśle zdominowana przez  $B$ , lecz nie przez  $C$ .



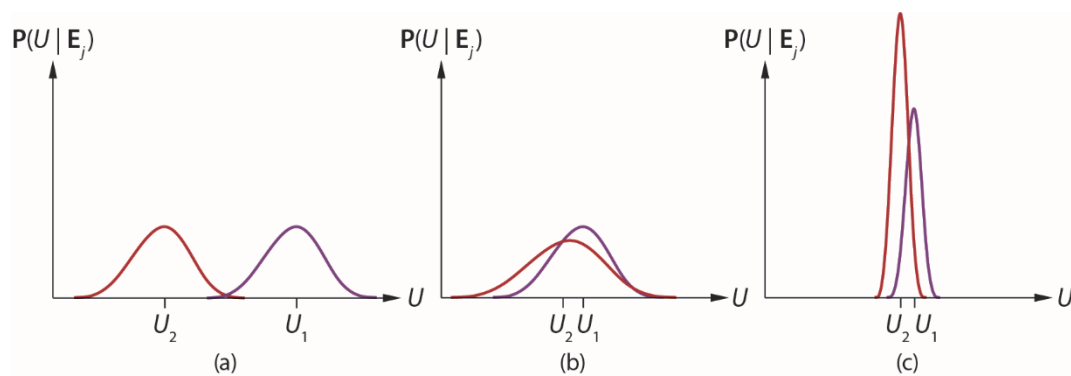
**RYSUNEK 16.5.** Dominacja stochastyczna. (a)  $S_2$  jest stochastycznie zdominowane przez  $S_1$  pod względem *Oszczędności* (ujemny koszt). (b) Skumulowane rozkłady (dystrybuanty) *Oszczędności* dla  $S_1$  i  $S_2$ .



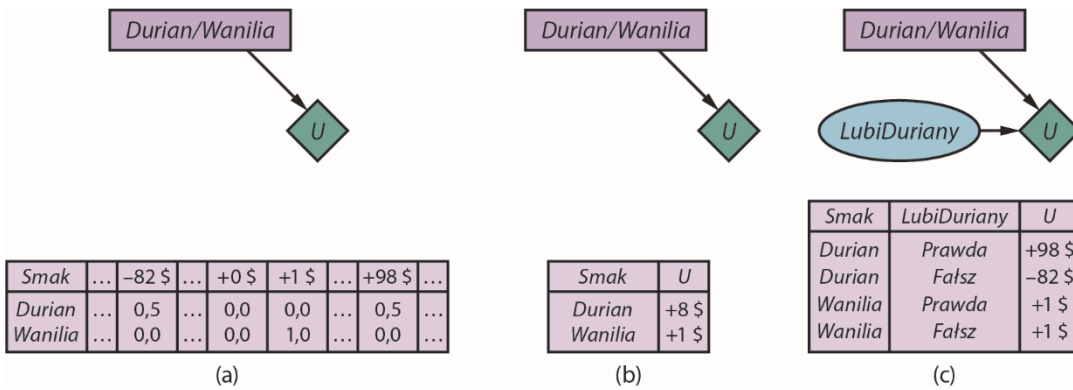
**RYSUNEK 16.6.** Sieć decyzyjna odpowiadająca problemowi wyboru lokalizacji portu lotniczego.



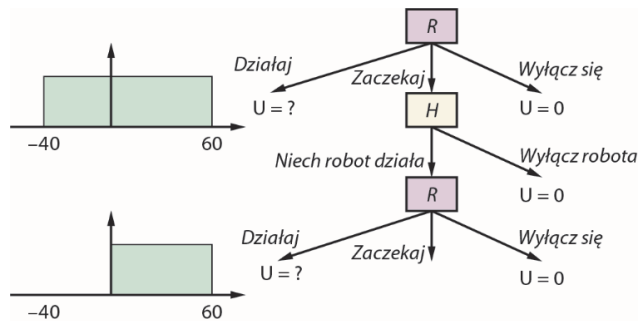
**RYSUNEK 16.7.** Uproszczona reprezentacja problemu lokalizacji portu lotniczego — pokazane są węzły losowe odpowiadające stanom końcowym.



**RYSUNEK 16.8.** Trzy generyczne przypadki wartości informacji. W przypadku (a) akcja  $a_1$  prawie na pewno góruje na  $a_2$ , zatem dodatkowa informacja nie jest potrzebna. W przypadku (b) wybór jest niejasny i dodatkowa informacja ma kluczowe znaczenie. W przypadku (c) wybór także jest nieoczywisty, lecz ze względu na niewielką różnicę między opcjami dodatkowa informacja ma niewielkie znaczenie. (Uwaga: fakt, że  $U_2$  ma wysoki pik w przypadku (c) oznacza, że jej wartość oczekiwana znana jest z większą pewnością niż wartość oczekiwana  $U_1$ ).



**RYСУNEK 16.9.** (a) Sieć decyzyjna dla problemu wyboru lodów w warunkach niepewnej funkcji użyteczności. (b) Sieć z zaznaczonymi wartościami oczekiwanej użyteczności dla każdej akcji. (c) Przeniesienie niepewności z funkcji użyteczności na nową zmienną losową.

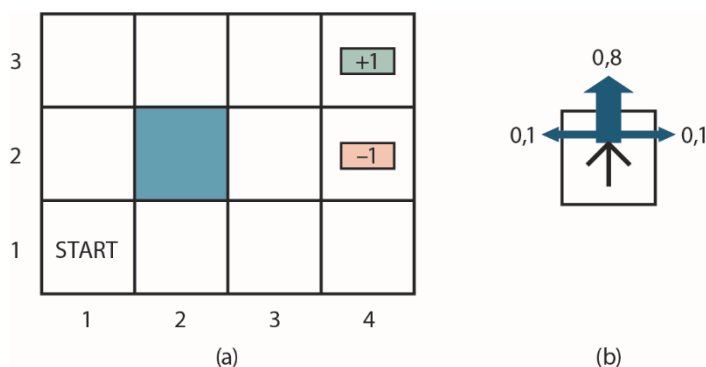


**RYСУNEK 16.10.** Lojalny robot *R*, wysoce niepewny preferencji człowieka *H*, może zdecydować o wyborze akcji bądź skonsultować swój zamiar z *H*. *H* może wyłączyć *R* albo pozwolić mu nadal działać. W tym drugim przypadku *R* ponownie staje przed takim samym wyborem; gdy postanowi działać, nadal jest niepewny wartości swej akcji dla *H*, ale ma gwarancję, że będzie to wartość nieujemna



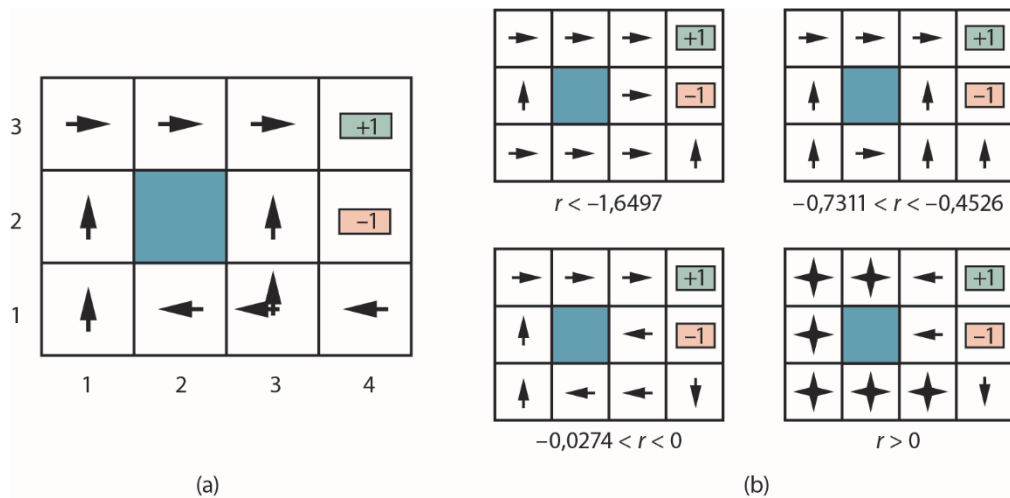
## ROZDZIAŁ 17

# PODEJMOWANIE ZŁOŻONYCH DECYZJI



**RYSUNEK 17.1.** (a) Proste środowisko stochastyczne, w którym działa agent rozwiązujący sekwencyjny problem decyzyjny. (b) Schemat modelu przejścia w środowisku: „zamierzona” akcja realizowana jest z prawdopodobieństwem 0,8, natomiast z prawdopodobieństwem 0,2 wykonywany jest ruch w kierunku prostopadłym do zamierzonego; w przypadku, gdy ruch jest niemożliwy z powodu natrafienia na przeszkodę (ścianę), agent pozostaje w dotychczasowym położeniu. Przejście do jednego ze stanów końcowych skutkuje nagrodą<sup>2</sup> w wysokości (odpowiednio) +1 albo -1, przejście do dowolnego stanu niekońcowego wiąże się z nagrodą w wysokości -0,04. Kwadrat (2, 2) w kolorze niebieskim otoczony jest ścianami ze wszystkich stron.

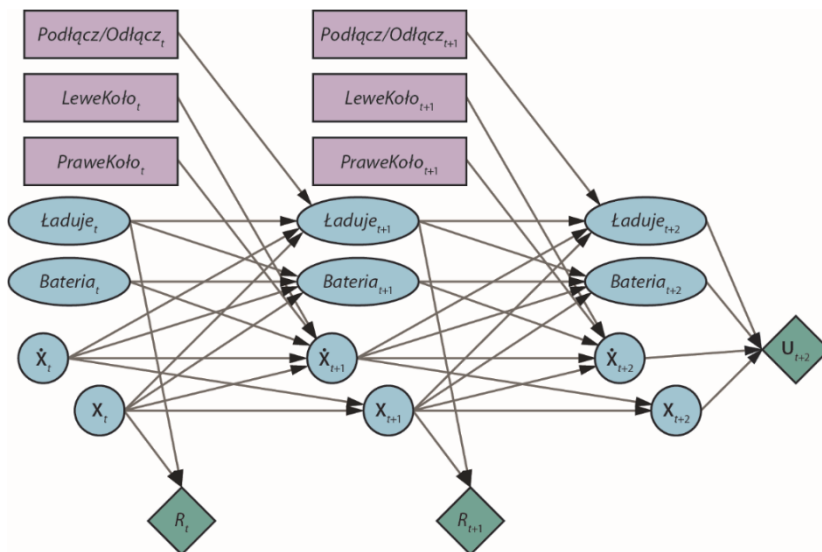
<sup>2</sup> W tym rozdziale pod pojęciem „nagrody” rozumiemy ilościową konsekwencję wykonania pewnej akcji, więc może być ona ujemna, wbrew intuicyjnemu rozumieniu tego słowa — *przyp. tłum.*



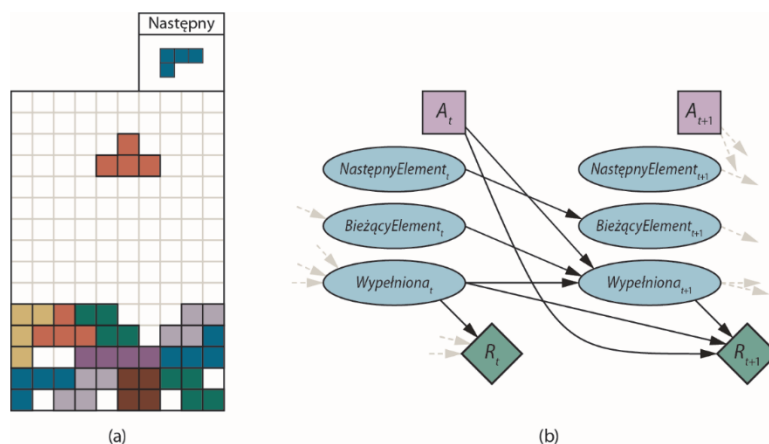
**RYSUNEK 17.2.** (a) Optymalne polityki dla środowiska stochastycznego z nagrodą  $r = -0,04$  dla przejścia między stanami niekończowymi. Są dwie takie polityki, ponieważ w stanie (3, 1) optymalne są obie akcje  $W\_górę$  i  $W\_lewo$ . (b) Optymalne polityki dla czterech różnych przedziałów wartości  $r$ .

3	0,8516	0,9078	0,9578	+1
2	0,8016		0,7003	-1
1	0,7453	0,6953	0,6514	0,4279
	1	2	3	4

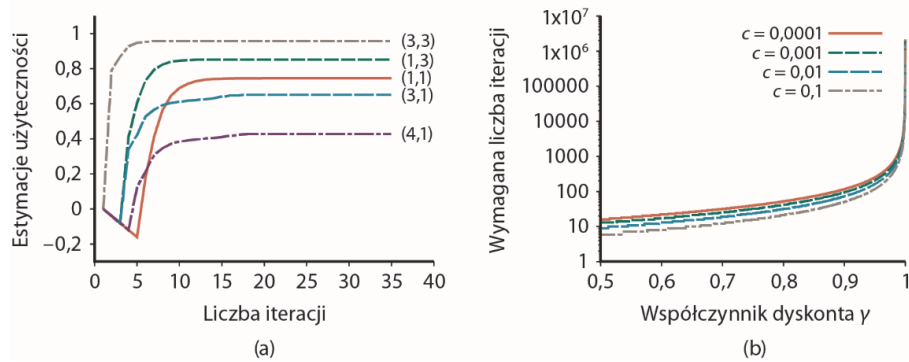
**RYSUNEK 17.3.** Użyteczności stanów środowiska z rysunku 17.1 dla współczynnika dyskonta  $\gamma = 1$  i nagrody  $r = -0,04$  dla przejścia między stanami niekończowymi.



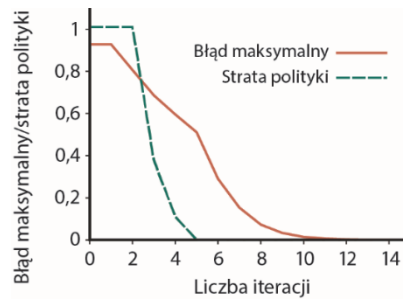
**RYСУNEK 17.4.** Dynamiczna sieć decyzyjna dla mobilnego robota ze zmiennymi stanu reprezentującymi stopień naładowania baterii, bieżącą lokalizację i bieżącą prędkość robota, oraz zmiennymi akcji reprezentującymi uruchamianie silników (lewego i prawego) i ładowanie baterii.



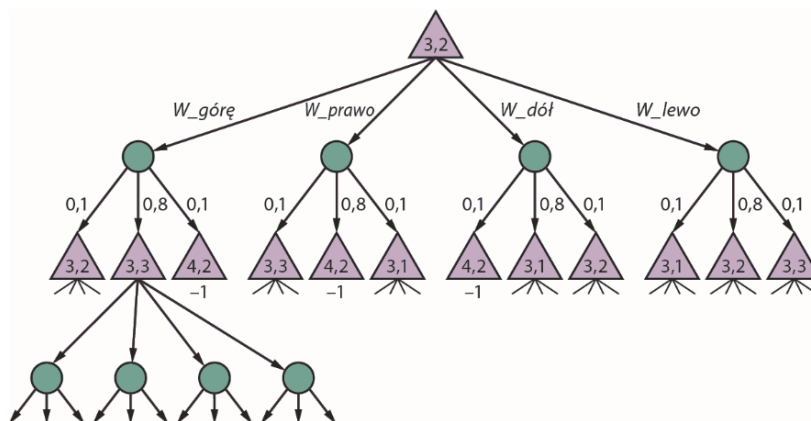
**RYСУNEK 17.5.** (a) Plansza gry Tetris. Element w kształcie litery T, znajdujący się pośrodku u góry, może być dowolnie obracany i przesuwany w poziomie, po czym upuszczony na zbiór elementów u dołu planszy. Gdy w wyniku tej operacji skompletowany zostanie poziomy rząd kwadratów, zostaje on usunięty z planszy, a elementy spoczywające powyżej niego spadają; agent zostaje wówczas nagrodzony jednym punktem. Widoczny dotychczas w prawym górnym narożniku następny element staje się elementem bieżącym, a w narożniku pojawia się nowy następny element, o kształcie wybranym losowo spośród siedmiu możliwych. Gra kończy się w momencie zapełnienia planszy i niemożności upuszczenia elementu. (b) Sieć DDN odpowiadająca grze Tetris w postaci problemu MDP.



**RYSUNEK 17.6.** (a) Wykres ilustrowujący ewolucję użyteczności wybranych stanów, wynikającą z algorytmu iterowania wartości. (b) Wykres wpływu współczynnika dyskonta  $\gamma$  na liczbę iteracji algorytmu iterowania wartości niezbędną do zagwarantowania, że wartość błędu nie przekracza  $c \cdot R_{\max}$ .



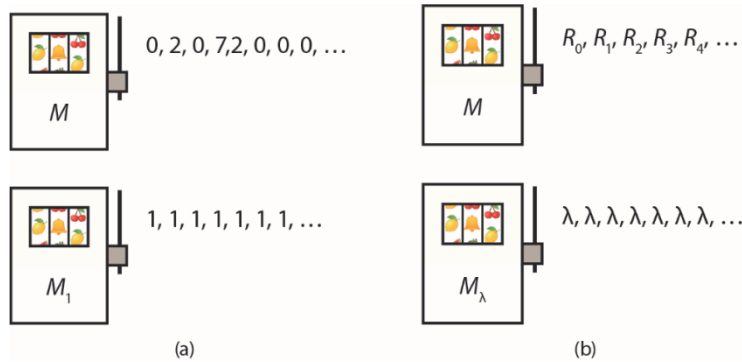
**RYSUNEK 17.7.** Błąd maksymalny  $\|U_t - U\|$  estymacji użyteczności oraz strata polityki  $\|U^{\pi_t} - U\|$  w funkcji liczby iteracji algorytmu iterowania wartości dla środowiska z rysunku 17.1.



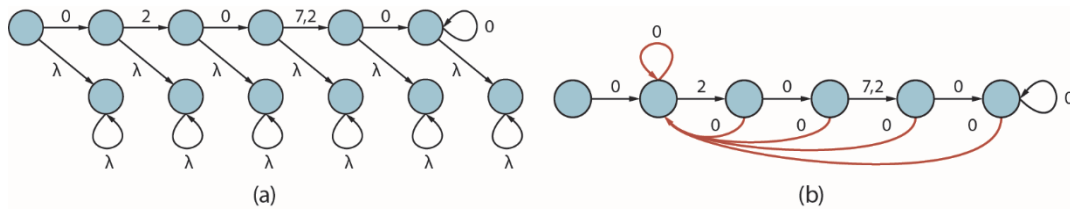
**RYСУNEK 17.8.** Fragment drzewa *expectimax* dla problemu MDP odpowiadającego środowisku z rysunku 17.1; korzeń drzewa odpowiada stanowi (3, 2). Trójkątne węzły są węzłami *max*, koliste węzły to węzły losowe.



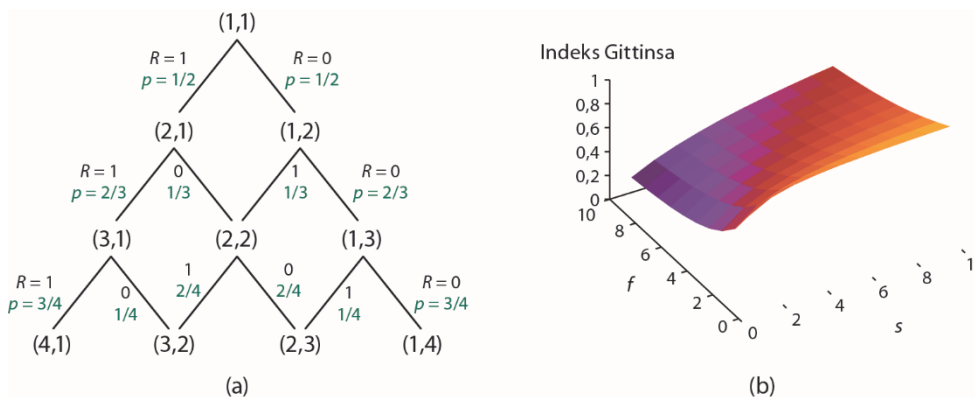
**RYСУNEK 17.9.** Uśredniona wydajność algorytmu UCT przypadająca na jeden punkt danych w funkcji liczby rozgrywek w przeliczeniu na jeden ruch, dla środowiska z rysunku 17.1. Wyniki uśrednione po 1000 uruchomieniach algorytmu.



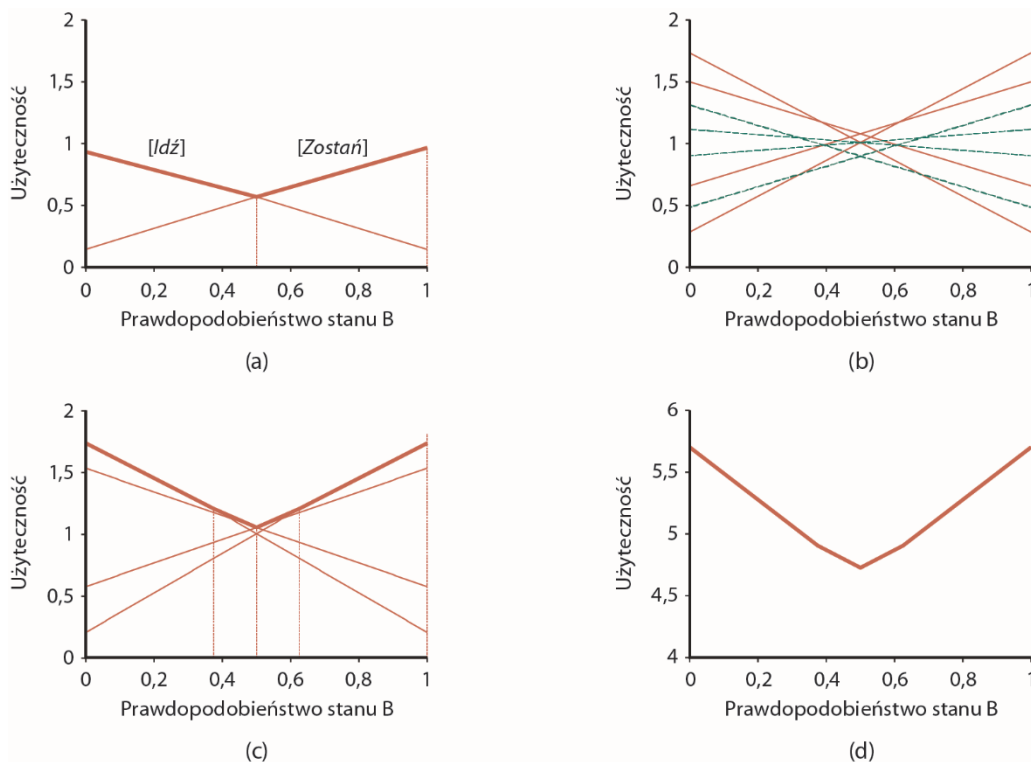
**RYSUNEK 17.10.** (a) Prosta wersja problemu bandyty z dwiema deterministycznymi dźwigniami. Dźwignie mogą być pociągane w dowolnej kolejności, bez wpływu na generowane nagrody. (b) Bardziej ogólny wariant problemu: pierwsza dźwignia generuje nieznaną a priori sekwencję nagród, druga daje ustaloną nagrodę  $\lambda$  przy każdym pociągnięciu.



**RYSUNEK 17.11.** (a) Sekwencja nagród  $M = 0, 2, 0, 7, 2, 0, 0, 0, \dots$  rozszerzona o wybór momentu permanentnego przełączenia się na ustaloną dźwignię  $M_\lambda$ . (b) Problem MDP, w którym optymalna wartość jest dokładnie równa optymalnej wartości dla (a) w punkcie przełączenia, w którym optymalna polityka nie rozróżnia  $M$  i  $M_\lambda$ .

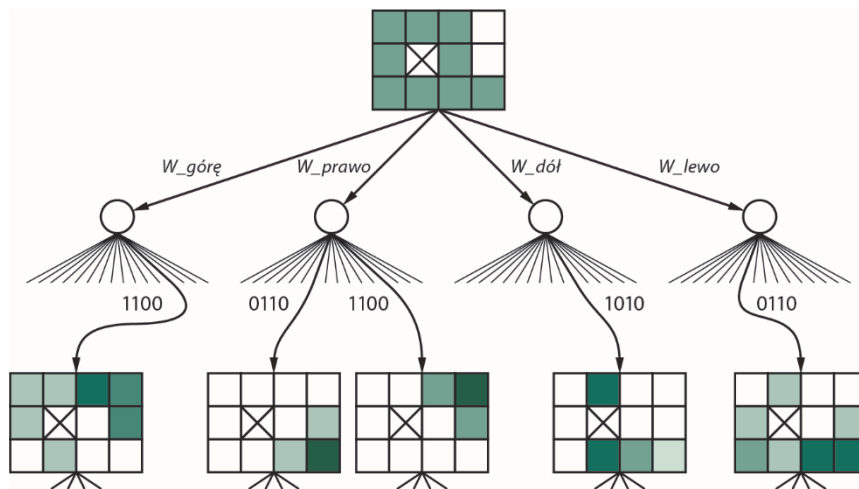


**RYSUNEK 17.12.** (a) Stany, nagrody i prawdopodobieństwa przejść dla problemu bandyty zero-jedynkowego. (b) Indeksy Gittinsa dla procesu bandyty zero-jedynkowego.

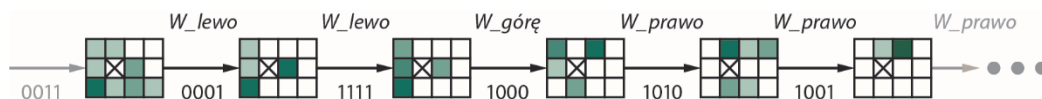


**RYСУNEK 17.13.** (a) Użyteczność dwóch jednokrokowych planów jako funkcja początkowego stanu przekonań  $b(B)$  dla świata z dwoma stanami  $A$  i  $B$ ; wykres odpowiedniej funkcji użyteczności wyróżniony został przez pogrubienie linii. (b) Użyteczności ośmiu różnych planów dwukrokowych. (c) Użyteczności czterech niezdominowanych planów dwukrokowych. (d) Funkcja użyteczności dla optymalnych planów 8-krokowych.





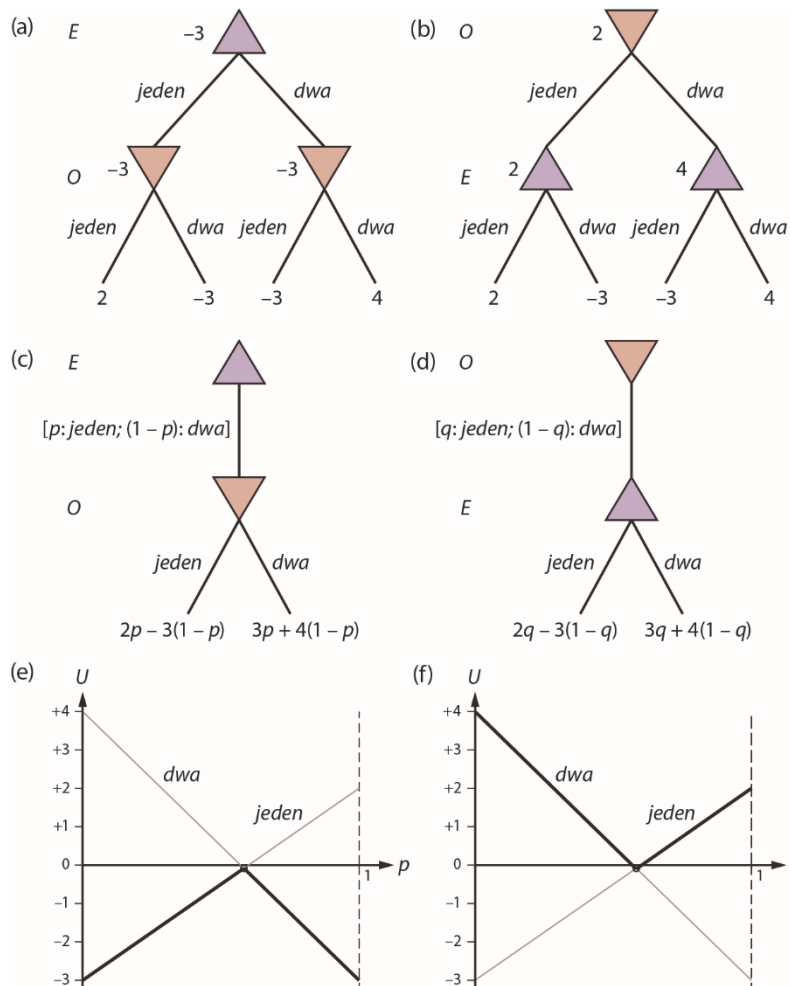
**RYСУNEK 17.14.** Fragment drzewa *expectimax* dla problemu POMDP odpowiadającego środowisku z rysunku 17.1, z równomiernie rozłożonymi prawdopodobieństwami w początkowym stanie przekonań. Zróżnicowana intensywność zacięniowania pól odpowiada zróżnicowanym prawdopodobieństwom znajdowania się agenta w danej lokalizacji.



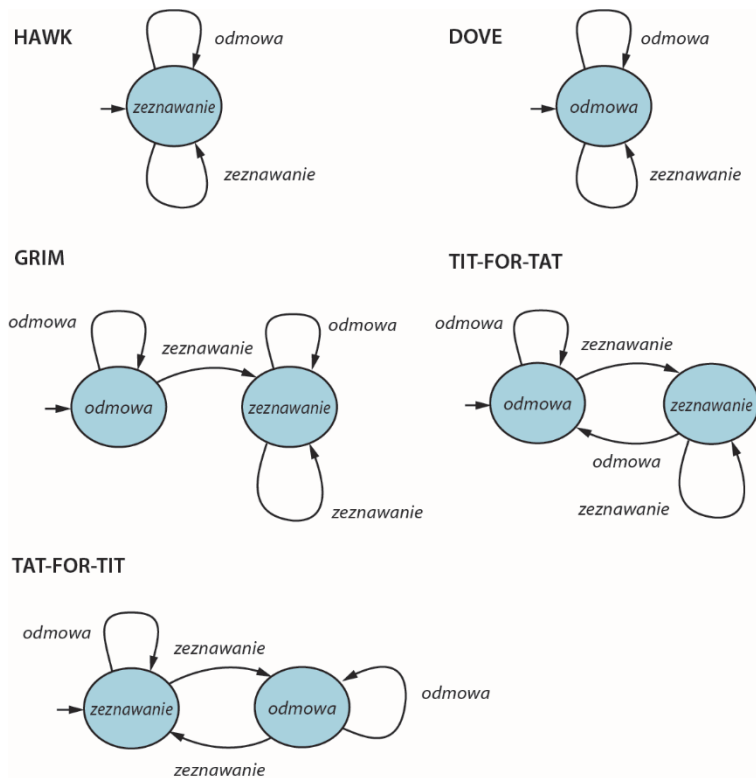
**RYСУNEK 17.15.** Sekwencja perceptów, stanów przekonań i akcji problemu POMDP odpowiadającego środowisku z rysunku 17.1, gdy czujnik wykrywający ścianę działa z prawdopodobieństwem błędu  $\epsilon = 0,2$ . Zauważmy, jak bezpieczne są początkowe ruchy *W\_lewo*: jest mało prawdopodobne zaprowadzenie agenta do stanu (4, 2), czyli stanu końcowego o wartości  $-1$ , a zbiór możliwych lokalizacji agenta redukuje się do niewielu pozycji. Po wykonaniu akcji *W\_góre* agent może przypuszczać, że znajduje się w stanie (3, 3), choć jest możliwe, że jego nowym stanem jest (1, 3). Na szczęście w obu przypadkach rozsądną decyzją jest akcja *W\_prawo*: agent przesuwa się prawo, wnioskuje, że przesunął się z (1, 3) do (2, 3), i akcja *W\_prawo* doprowadza go do stanu końcowego  $+1$ .

## ROZDZIAŁ 18

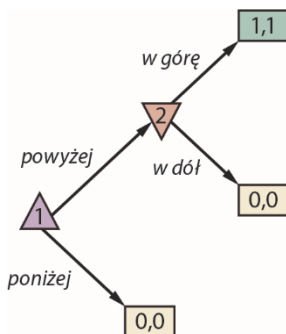
# PODEJMOWANIE DECYZJI W ŚRODOWISKU WIELOAGENTOWYM



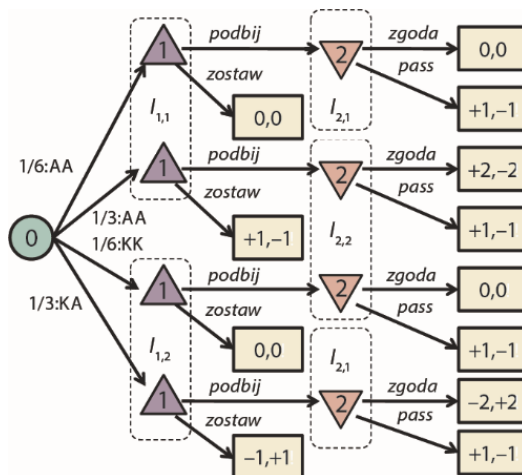
**RYSUNEK 18.1.** (a) i (b) Drzewa minimaksowe gry „parzyste-nieparzyste” w sytuacji, gdy gracze stosują czyste strategie. (c) i (d) Parametryzowane drzewa gry w sytuacji, gdy pierwszy gracz stosuje strategię mieszaną. Wartości wypłat zależne są od parametrów prawdopodobieństwa ( $p$  i  $q$ ) tej strategii. (e) i (f) Dla określonych wartości parametrów prawdopodobieństwa drugi gracz wybierze „lepszą” z dwóch akcji. Pogrubione linie ukazują wartość strategii mieszanej pierwszego gracza, który powinien wybrać dla swej strategii mieszanej parametry prawdopodobieństwa odpowiadające punktowi przecięcia tych linii.



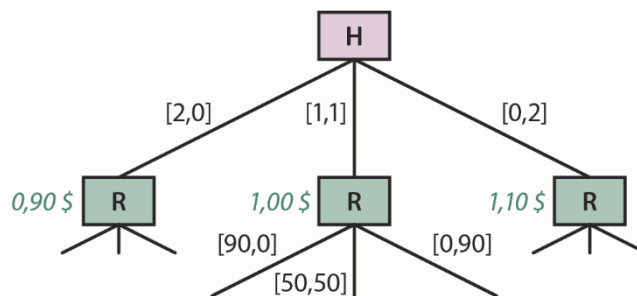
**RYСУNEK 18.2.** Maszyny skończenie stanowe reprezentujące wybrane strategie dla nieskończenie powtarzanej gry w dylemat więźnia.



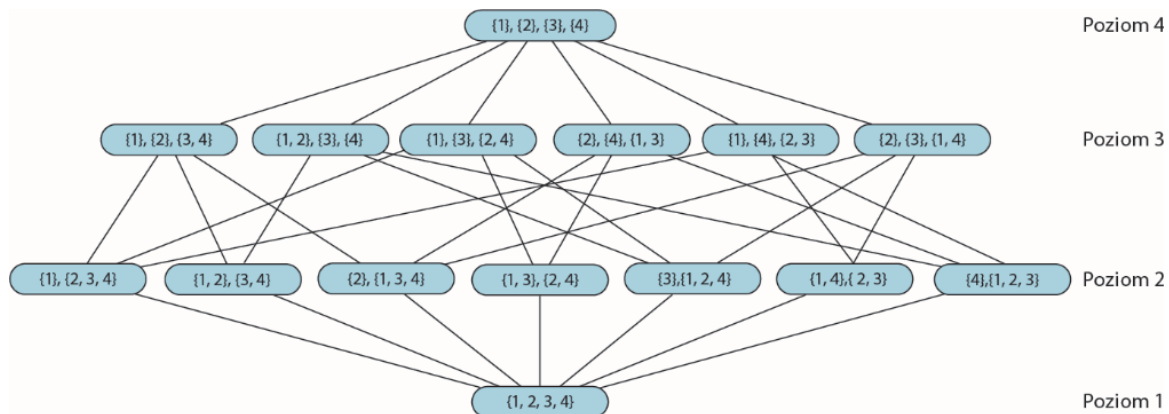
**RYСУNEK 18.3.** Gra w postaci ekstensywnej ilustrująca nieintuicyjną równowagę Nasha.



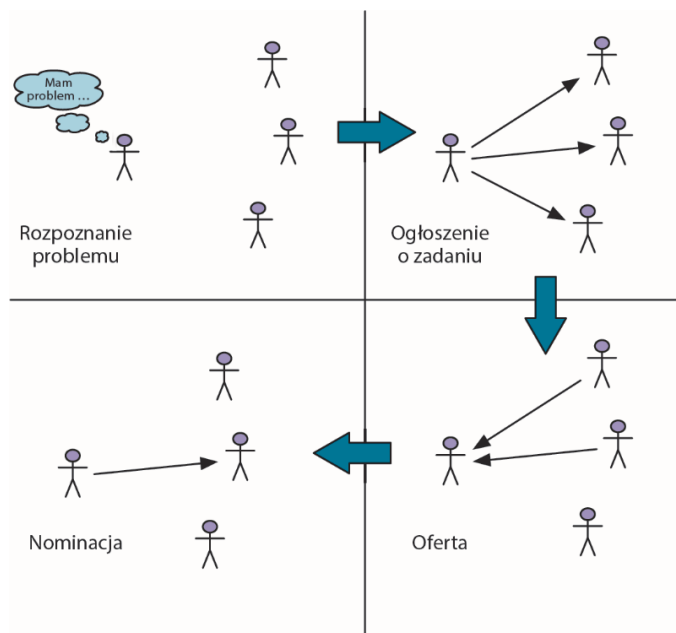
RYSUNEK 18.4. Ekstensywna postać gry w uproszczonej wersji pokera (dwóch graczy i cztery karty).



RYSUNEK 18.5. Gra w spinacze. Każda gałąź etykietowana jest parą liczb całkowitych, z których pierwsza odnosi się do spinaczy, a druga do zszywek wyprodukowanych dla danego wyboru. Harriet ma trzy warianty wyboru: dwa spinacze, dwie zszywki albo spinacz i zszywkę (kursywą wyróżniliśmy wartości gry dla Harriet, gdyby gra kończyła się w tym miejscu; przyjęliśmy  $\theta = 0,45$ ). Robbie ma następnie trzy warianty wyboru: wyprodukowanie 90 spinaczy, wyprodukowanie 90 zszywek albo wyprodukowanie 50 spinaczy i 50 zszywek.



RYSUNEK 18.6. Graf struktury koalicji dla zbioru graczy  $N = \{1, 2, 3, 4\}$ . Na poziomie 1 reprezentowana jest pojedyncza (wielka) koalicja, na poziomie 2 możliwe są podziały na dwie koalicje, itd.



RYSUNEK 18.7. Przydział zadań według protokołu sieci kontraktowej.