

## Logika inteligencji

Czy istnieje „esencja inteligencji”, która odróżnia inteligentne systemy od nieinteligentnych? Jeśli tak, to czym ona jest? Ten rozdział sugeruje odpowiedź na te pytania, wprowadzając idee stojące za projektem NARS (Nonaxiomatic Reasoning System). NARS opiera się na opinii, że istotą inteligencji jest zdolność do adaptacji przy niewystarczającej wiedzy i zasobach. Zgodnie z tym przekonaniem autor zaprojektował nową formalną logikę i zaimplementował ją w systemie komputerowym. Taka „logika inteligencji” dostarcza ujednoczonego wyjaśnienia wielu funkcji poznawczych ludzkiego umysłu i jest również wystarczająco konkretna, aby pokierować faktyczną budową „myślącej maszyny” ogólnego przeznaczenia.

## Inteligencja i logika

### Definicja inteligencji

Debata na temat istoty inteligencji trwa od dziesięcioleci, ale nadal nie ma oznak konsensusu (sama ta książka jest tego dowodem). W „głównym nurcie sztucznej inteligencji” przedstawiono kilka reprezentatywnych opinii:

„AI zajmuje się metodami osiągania celów w sytuacjach, w których dostępne informacje mają pewien złożony charakter.

Metody, które muszą zostać użyte, są związane z problemem przedstawionym w sytuacji i są podobne, niezależnie od tego, czy rozwiązujący problem jest człowiekiem, Marsjaninem czy programem komputerowym”.

Inteligencja zazwyczaj oznacza „zdolność do rozwiązywania trudnych problemów”.

„Przez „ogólne inteligentne działanie” chcemy wskazać ten sam zakres inteligencji, jaki widzimy w działaniu człowieka: że w każdej rzeczywistej sytuacji zachowanie odpowiednie do celów systemu i adaptacyjne do wymagań środowiska może wystąpić, w pewnych granicach szybkości i złożoności”.

Być może jest za wcześnie, aby zdefiniować inteligencję. Oczywiście jest, że po dziesięcioleciach badań nadal niewiele o niej wiemy. Jest więcej pytań niż odpowiedzi. Każda definicja oparta na obecnej wiedzy skazana jest na rewizję przez przyszłe prace. Wszyscy wiemy, że dobrze uzasadniona definicja jest zazwyczaj wynikiem, a nie punktem wyjścia, badań naukowych. Jednak nadal istnieją powody, dla których powinniśmy być zaniepokojeni definicją inteligencji w obecnych czasach. Chociaż wyjaśnienie znaczenia pojęcia zawsze pomaga w komunikacji, problem ten jest szczególnie ważny w przypadku AI. Jako społeczność, badacze AI muszą uzasadnić swoją dziedzinę jako dyscyplinę naukową. Bez (stosunkowo) jasnej definicji inteligencji trudno powiedzieć, dlaczego AI różni się na przykład od informatyki lub psychologii. Czy jest coś nowego i wyjątkowego, czy też tylko wymyślne etykiety na starych rzeczach? Co ważniejsze, każdy badacz w tej dziedzinie musi uzasadnić swój plan badawczy zgodnie z taką definicją. Każdy, kto chce pracować nad sztuczną inteligencją, staje przed dwuetapowym zadaniem: wybrać roboczą definicję inteligencji, a następnie stworzyć ją na komputerze. Robocza definicja to definicja na tyle konkretna, że można z nią bezpośrednio pracować. Akceptując roboczą definicję inteligencji, nie oznacza to, że naprawdę wierzysz, że w pełni oddaje ona pojęcie „inteligencji”, ale że potraktujesz ją jako cel swojego obecnego projektu badawczego. Dlatego brak konsensusu co do tego, czym jest inteligencja, nie uniemożliwia każdemu badaczowi wybrania (świadomie lub nie) roboczej definicji inteligencji. Właściwie, dopóki nie zachowasz jednej (lub więcej niż jednej) definicji, nie możesz twierdzić, że pracujesz nad sztuczną inteligencją. Akceptując roboczą definicję inteligencji, najważniejsze zobowiązania, jakie podejmuje badacz, dotyczą dopuszczalnych założeń i pożądaných wyników, które wiążą całą konkretną pracę, która następuje. Wady definicji trudno zrekompensować

badaniami, a niewłaściwe definicje utrudnią badania bardziej niż to konieczne lub odwiodą je od pierwotnego celu. Przed zbadaniem konkretnych roboczych definicji inteligencji musimy ustalić ogólny standard tego, co sprawia, że definicja jest lepsza od innych. Carnap napotkał ten sam problem, gdy próbował wyjaśnić pojęcie „prawdopodobieństwo”. Zadanie „polega na przekształceniu danego, mniej lub bardziej niedokładnego pojęcia w dokładne lub raczej na zastąpieniu pierwszego drugim”, gdzie pierwsze może należeć do języka codziennego lub do poprzedniego etapu języka naukowego, a drugie musi być podane przez wyraźne reguły jego użycia. Według Carnapa, drugie pojęcie, lub definicja robocza, jak jest nazywana w tym rozdziale, musi spełniać następujące wymagania:

1. Jest podobne do pojęcia, które ma być zdefiniowane, na ile pozwala na to jego niejasność.
2. Jest zdefiniowane w dokładnej formie.
3. Jest owocne w badaniu.
4. Jest proste, na ile pozwalają inne wymagania.

Wydaje się, że te wymagania są również rozsądne i odpowiednie dla naszego obecnego celu. Teraz zobaczymy, co konkretnie oznaczają dla definicji roboczej inteligencji:

Podobieństwo (do standardowego użycia). Chociaż „inteligencja” nie ma dokładnego znaczenia w języku potocznym, ma pewne powszechne zastosowania, z którymi definicja robocza powinna się zgadzać. Na przykład, zwykli ludzie są inteligentni, ale większość zwierząt i maszyn (w tym zwykłe systemy komputerowe) albo wcale nie jest inteligentna, albo jest znacznie mniej inteligentna od ludzi.

Dokładność (lub dobrze zdefiniowana). Biorąc pod uwagę definicję roboczą, to, czy (lub w jakim stopniu) system jest inteligentny, powinno być jasno rozstrzygalne. Z tego powodu inteligencji nie można definiować w kategoriach innych źle zdefiniowanych pojęć, takich jak umysł, myślenie, poznanie, intencjonalność, racjonalność, mądrość, świadomość itd., chociaż pojęcia te mają bliskie związki z inteligencją.

Owocność (i pouczanie). Definicja robocza powinna zawierać konkretne wytyczne dla badań na niej opartych – na przykład, jakie założenia można zaakceptować, jakie zjawiska można zignorować, jakie właściwości są pożądane itd. Co najważniejsze, definicja robocza inteligencji powinna przyczyniać się do rozwiązywania podstawowych problemów w AI.

Prostota. Chociaż inteligencja jest z pewnością złożonym mechanizmem, robocza definicja powinna być prosta. Z teoretycznego punktu widzenia prosta definicja umożliwi szczegółowe zbadanie teorii; z praktycznego punktu widzenia prosta definicja jest łatwa w użyciu.

Dla naszych obecnych celów nie ma „właściwej” ani „niewłaściwej” roboczej definicji inteligencji, ale są „lepsze” i „niezbyt dobre”. Porównując proponowane definicje, cztery wymagania mogą być ze sobą sprzeczne. Na przykład jedna definicja jest bardziej owocna, podczas gdy inna jest prostsza. W takiej sytuacji konieczne staje się pewne ważenie i kompromis. Nie ma jednak dowodów wskazujących na to, że ogólnie rzecz biorąc, wymagania nie mogą być spełnione w tym samym czasie.

### **Robocza definicja inteligencji**

Po przygotowaniu poprzedniej sekcji proponujemy tutaj roboczą definicję inteligencji:

Inteligencja to zdolność systemu do adaptacji do swojego otoczenia, podczas gdy działa on z niewystarczającą wiedzą i zasobami.

Środowiskiem systemu może być świat fizyczny lub inne systemy przetwarzania informacji (ludzkie lub komputerowe). W obu przypadkach interakcje można opisać doświadczeniami (lub bodźcami) i reakcjami systemu, które są odpowiednio strumieniami informacji wejściowych i wyjściowych. Dla systemu postrzegalne wzorce danych wejściowych i wytwarzalne wzorce danych wyjściowych stanowią jego język interfejsu. Adaptacja oznacza, że system uczy się na podstawie swoich doświadczeń. Dostosowuje swoją wewnętrzną strukturę, aby zbliżyć się do swoich celów, tak jakby przyszłe sytuacje były podobne do sytuacji z przeszłości. Nie wszystkie systemy dostosowują się do swojego otoczenia. Na przykład tradycyjny system komputerowy uzyskuje całą swoją wiedzę w fazie projektowania. Później jego doświadczenie nie przyczynia się do jego zachowań. Aby zdobyć nową wiedzę, taki system musiałby zostać przeprojektowany. Niewystarczająca wiedza i zasoby oznaczają, że system działa w ramach następujących ograniczeń:

Skończone. System ma stałą zdolność przetwarzania informacji.

W czasie rzeczywistym. Wszystkie zadania mają przypisane wymagania czasowe.

Otwarte. Nie ma ograniczeń co do wiedzy i zadań, które system może zaakceptować, o ile są one możliwe do przedstawienia w języku interfejsu.

Dwa główne składniki definicji roboczej, adaptacja oraz niewystarczająca wiedza i zasoby, są ze sobą powiązane. System adaptacyjny musi mieć pewne niedobory wiedzy i zasobów, w przeciwnym razie nigdy nie musiałby się zmieniać. Z drugiej strony, bez adaptacji system może mieć niewystarczającą wiedzę i zasoby, ale nie podejmować żadnych prób poprawy swoich możliwości. Nie wszystkie systemy w pełni biorą pod uwagę własne niedobory wiedzy i zasobów. Na przykład systemy nieadaptacyjne po prostu ignorują nową wiedzę w swoich interakcjach ze swoim otoczeniem. Jeśli chodzi o sztuczne systemy adaptacyjne, większość z nich nie jest skończona, w czasie rzeczywistym i otwarta w następujących znaczeniach:

1. Chociaż wszystkie rzeczywiste systemy są skończone, wiele modeli teoretycznych (na przykład maszyna Turinga) pomija fakt, że wymagania dotyczące czasu procesora i/lub przestrzeni pamięci mogą wykraczać poza możliwości systemu.

2. Większość obecnych systemów AI nie bierze pod uwagę ograniczeń czasowych w czasie wykonywania. Większość systemów czasu rzeczywistego może obsłużyć ograniczenia czasowe tylko wtedy, gdy są one zasadniczo terminami [35].

3. Na to, czego system może doświadczyć, nakładane są różne ograniczenia. Na przykład dopuszczalne są tylko pytania, na które można odpowiedzieć poprzez wyszukiwanie i dedukcję z bieżącej wiedzy, nowa wiedza nie może być w konflikcie z poprzednią wiedzą itd.

Wiele systemów komputerowych jest projektowanych z założeniem, że ich wiedza i zasoby, choć ograniczone lub ograniczone, są nadal wystarczające do wykonania zadań, które zostaną im powierzone. W sytuacji, gdy to założenie zawodzi, taki system po prostu wpada w panikę lub ulega awarii i prosi o zewnętrzną interwencję użytkownika. Aby system działał przy założeniu niewystarczającej wiedzy i zasobów, powinien mieć mechanizmy do obsługi następujących typów sytuacji, między innymi:

- nowy procesor jest wymagany, gdy wszystkie istniejące procesory są zajęte;
- dodatkowa pamięć jest wymagana, gdy cała dostępna pamięć jest już pełna;
- zadanie pojawia się, gdy system jest zajęty czymś innym;

- zadanie pojawia się z wymaganiem czasowym, więc wyczerpujące wyszukiwanie nie jest opcją;
- nowa wiedza koliduje z poprzednią wiedzą;
- zadano pytanie, na które nie można wywnioskować pewnej odpowiedzi z dostępnej wiedzy.

W przypadku tradycyjnych systemów komputerowych tego typu sytuacje zazwyczaj wymagają interwencji człowieka lub po prostu powodują, że system odmawia przyjęcia zadania lub wiedzy. Jednakże w przypadku systemu zaprojektowanego przy założeniu niewystarczającej wiedzy i zasobów są to normalne sytuacje i powinny być płynnie zarządzane przez sam system. Zgodnie z powyższą definicją inteligencja jest „wysoko rozwiniętą formą adaptacji umysłowej” [26]. Definiując inteligencję, wielu autorów ignoruje uzupełniające pytanie: co jest nieinteligentne? Jeśli wszystko jest inteligentne, to ta koncepcja jest pusta. Nawet jeśli zgodzimy się, że inteligencja, podobnie jak prawie wszystkie właściwości, jest kwestią stopnia, nadal potrzebujemy kryteriów, aby wskazać, co sprawia, że system jest bardziej inteligentny od innego. Ponadto, aby SI była (niezależną) dyscypliną, wymagamy, aby koncepcja „inteligencji” różniła się od innych ustalonych koncepcji, ponieważ w przeciwnym razie mówimy tylko o pewnych dobrze znanych rzeczach z nową nazwą, co nie wystarczy, aby ustanowić nową gałąź nauki. Na przykład, jeśli każdy system komputerowy jest inteligentny, lepiej pozostać w teorii obliczeń. Intuicyjnie „inteligentny system” nie oznacza szybszego i większego komputera. Z drugiej strony, nieinteligentny system niekoniecznie jest niezdolny lub daje tylko błędne wyniki. W rzeczywistości większość zwykłych systemów komputerowych i wiele zwierząt potrafi zrobić coś, czego ludzie nie potrafią. Jednak te zdolności nie dają im tytułu „inteligentnych”. Czego brakuje w tych zdolnych, ale nieinteligentnych systemach? Zgodnie z roboczą definicją inteligencji przedstawioną wcześniej, nieinteligentny system to taki, który nie dostosowuje się do swojego otoczenia. Szczególnie w systemach sztucznych nieinteligentny system to taki, który został zaprojektowany z założeniem, że działa tylko nad problemami, dla których system ma wystarczającą wiedzę i zasoby. Inteligentny system nie zawsze jest „lepszy” od nieinteligentnego systemu w praktycznych celach. W rzeczywistości jest odwrotnie: gdy problem może zostać rozwiązany przez oba, nieinteligentny system jest zwykle lepszy, ponieważ gwarantuje poprawne rozwiązanie. Jak powiedział Hofstadter, w przypadku zadań takich jak dodawanie dwóch liczb, „niezawodny, ale bezmyślny” system jest lepszy niż „inteligentny, ale omylny” system.

### **Porównanie z innymi definicjami**

Ponieważ nie można porównać powyższej definicji z każdą z istniejących definicji inteligencji, pogrupujemy je w kilka kategorii. Mówiąc ogólnie, badania nad sztuczną inteligencją mają dwie główne motywacje. Jako dziedzina nauki chcemy dowiedzieć się, jak działa ludzki umysł i „umysł” w ogóle; a jako gałąź technologii chcemy stosować komputery w dziedzinach, w których obecnie dobrze działa tylko ludzki umysł. Intuicyjnie oba cele można osiągnąć, jeśli uda nam się zbudować systemy komputerowe, które są „podobne do ludzkiego umysłu”. Ale w jakim sensie są „podobne”? Dla różnych osób pożądane podobieństwo może obejmować strukturę, zachowanie, pojemność, funkcję lub zasadę. W dalszej części omawiamy typowe opinie w każdej z pięciu kategorii, aby zobaczyć, dokąd te definicje inteligencji doprowadzą AI.

### **Aby symulować ludzki mózg**

Inteligencja jest wytwarzana przez ludzki mózg, więc być może AI powinna spróbować symulować mózg w systemie komputerowym tak wiernie, jak to możliwe. Taką opinię w skrajnej formie przedstawiają neurobiolodzy Reeke i Edelman, którzy twierdzą, że „ostateczne cele AI i neuronauki są dość podobne”. Chociaż utożsamianie AI z modelem mózgu wydaje się rozsądne, niewielu badaczy AI przyjmuje takie podejście w ścisłym tego słowa znaczeniu. Nawet ruch „sieci neuronowych” „nie

koncentruje się na modelowaniu neuronowym (tj. modelowaniu neuronów), ale raczej... koncentruje się na neuronalnie inspirowanym modelowaniu procesów poznawczych". Dlaczego? Jednym z oczywistych powodów jest zniechęcająca złożoność tego podejścia. Obecna technologia wciąż nie jest wystarczająco wydajna, aby symulować ogromną sieć neuronową, nie wspominając o fakcie, że mózg wciąż pozostaje tajemnicą. Co więcej, nawet gdybyśmy byli w stanie zbudować model mózgu na poziomie neuronów z dowolną pożądaną dokładnością, nie można by tego nazwać sukcesem AI, chociaż byłby to sukces neuronauki. AI jest bliżej związana z koncepcją „modelu umysłu” – to znaczy, wysokiego poziomu opisu aktywności mózgu, w którym nie pojawiają się koncepcje biologiczne. Preferowany jest opis wysokiego poziomu, nie dlatego, że opis niskiego poziomu jest niemożliwy, ale dlatego, że jest on zazwyczaj prostszy i bardziej ogólny. Charakterystyczną cechą AI jest próba „uzyskania umysłu bez mózgu” – to znaczy, opisanie umysłu w sposób niezależny od medium. Dotyczy to wszystkich modeli: budując model, koncentrujemy się na pewnych właściwościach obiektu lub procesu i ignorujemy nieistotne aspekty; robiąc tak, uzyskujemy spostrzeżenia, które są trudne do rozróżnienia w samym obiekcie lub procesie. Z tego powodu dokładna duplikacja nie jest modelem, a model zawierający niepotrzebne szczegóły nie jest dobrym modelem. Jeśli zgodzimy się, że „mózg” i „umysł” to różne koncepcje, to dobry model mózgu nie jest dobrym modelem umysłu, chociaż pierwszy jest przydatny sam w sobie i pomocny w budowaniu drugiego.

### **Aby powielać ludzkie zachowania**

Biorąc pod uwagę, że zawsze oceniamy inteligencję innych ludzi na podstawie ich zachowania, naturalne jest, że celem SI jest „odtworzenie zachowania wytworzonego przez ludzki umysł tak dokładnie, jak to możliwe”. Taka robocza definicja inteligencji wymaga od badaczy, aby używali testu Turinga [36] jako wystarczającego i koniecznego warunku posiadania inteligencji i aby poważnie traktowali dowody psychologiczne. Ze względu na naturę testu Turinga i ograniczenia zasobów konkretnego systemu komputerowego, nie ma mowy o tym, aby system miał z góry zapisane w swojej pamięci wszystkie możliwe pytania i właściwe odpowiedzi, a następnie przekonująco naśladował człowieka, przeszukując taką listę. Jedynym realistycznym sposobem naśladowania ludzkiego zachowania w rozmowie jest generowanie odpowiedzi w czasie rzeczywistym. Aby to zrobić, potrzebne są nie tylko zdolności poznawcze, ale także znacznie wcześniejsze „ludzkie doświadczenie” [9]. Dlatego musi mieć ciało, które czuje się jak człowiek, musi mieć wszystkie ludzkie motywacje (w tym biologiczne) i musi być traktowane przez ludzi jak istota ludzka – więc musi być po prostu „sztucznym człowiekiem”, a nie systemem komputerowym ze sztuczną inteligencją. Jak zauważa French, wykorzystując zachowanie jako dowód, test Turinga jest kryterium wyłącznie dla ludzkiej inteligencji, a nie dla inteligencji w ogóle [9]. Takie podejście może prowadzić do dobrych modeli psychologicznych, które są cenne z wielu powodów, ale cierpi na „ludzki szowinizm” [13] – musielibyśmy powiedzieć, zgodnie z definicją, że science-fictionowy obcy stwór E. T. nie był inteligentny, ponieważ z pewnością nie zdałby testu Turinga. Chociaż „odtworzenie ludzkiego (werbalnego) zachowania” może być nadal wystarczającym warunkiem bycia inteligentnym (jak sugeruje Turing), taki cel jest trudny, jeśli nie niemożliwy do osiągnięcia. Co ważniejsze, nie jest to warunek konieczny bycia inteligentnym, jeśli chcemy, aby „inteligencja” była bardziej ogólnym pojęciem niż „inteligencja ludzka”.

### **Rozwiązywanie trudnych problemów**

W języku potocznym „inteligentny” jest zwykle stosowany do osób, które potrafią rozwiązywać trudne problemy. Zgodnie z tym typem definicji inteligencja to zdolność rozwiązywania trudnych problemów, a sposób rozwiązywania problemów nie ma większego znaczenia. Jakie problemy są „trudne”? Na początku rozwoju sztucznej inteligencji wielu badaczy zajmowało się czynnościami intelektualnymi, takimi jak granie w gry i dowodzenie twierdzeń. Obecnie twórcy systemów eksperckich koncentrują się na „problemach ze świata rzeczywistego”, które pojawiają się w różnych domenach. Założenie

stojące za tym podejściem jest następujące: „Oczywiście eksperci są inteligentni, więc jeśli system komputerowy może rozwiązywać problemy, które tylko eksperci potrafią rozwiązać, system komputerowy również musi być inteligentny”. Dlatego wiele osób uważa sukces komputera grającego w szachy Deep Blue za sukces sztucznej inteligencji. Ruch ten przyciągnął wielu badaczy, wyprodukował wiele praktycznie użytecznych systemów, przyciągnął znaczne fundusze i w ten sposób wniósł istotny wkład w rozwój przedsiębiorstwa zajmującego się sztuczną inteligencją. Zazwyczaj systemy są opracowywane poprzez analizę wiedzy domenowej i strategii eksperckiej, a następnie wbudowywanie ich w system komputerowy. Jednakże, choć często dochodowe, systemy te nie dają zbyt wielu informacji na temat tego, jak działa umysł. Nic dziwnego, że ludzie pytają, po zapoznaniu się z działaniem takiego systemu, „Gdzie jest SI?” – systemy te wyglądają jak zwykłe systemy aplikacji komputerowych i nadal cierpią z powodu dużej sztywności i kruchości (czego SI chce uniknąć). Jeśli inteligencję definiuje się jako „zdolność do rozwiązywania trudnych problemów”, to następne pytanie brzmi: „Trudne dla kogo?” Jeśli powiemy „trudne dla ludzi”, to większość istniejącego oprogramowania komputerowego jest już inteligentna – żaden człowiek nie potrafi zarządzać bazą danych tak dobrze, jak systemem zarządzania bazą danych, ani podmieniać słów w pliku tak szybko, jak program do edycji. Jeśli powiemy „trudne dla komputerów”, to SI staje się „czymś, czego jeszcze nie zrobiono”, co zostało nazwane „twierdzeniem Teslera”. Pogląd, że SI jest „nieustannie rozszerzającą się granicą”, czyni ją atrakcyjną i ekscytującą, na co zasługuje, ale niewiele nam mówi o tym, jak różni się od innych obszarów badań w informatyce – czy można uczciwie powiedzieć, że problemy tam występujące są łatwe? Jeśli badacze sztucznej inteligencji nie będą w stanie zidentyfikować innych wspólnych cech problemów, którymi się zajmują, poza samą trudnością, mało prawdopodobne jest, aby poczynili jakiegokolwiek postępy w zrozumieniu i powielaniu inteligencji.

### **Wykonywanie funkcji poznawczych**

Zgodnie z tym poglądem inteligencję charakteryzuje zestaw funkcji poznawczych, takich jak rozumowanie, percepcja, pamięć, rozwiązywanie problemów, używanie języka itd. Badacze, którzy podzielają ten pogląd, zazwyczaj koncentrują się tylko na jednej z tych funkcji, opierając się na idei, że badania nad wszystkimi funkcjami będą mogły zostać ostatecznie połączone w przyszłości, aby uzyskać pełny obraz inteligencji. „Funkcja poznawcza” jest często definiowana w sposób ogólny i abstrakcyjny. To podejście wytworzyło i będzie nadal wytwarzać narzędzia w postaci pakietów oprogramowania, a nawet specjalistycznego sprzętu, z których każdy może wykonywać funkcję podobną do pewnych umiejętności umysłowych ludzi, a zatem może być używany w różnych domenach w celach praktycznych. Jednak tego rodzaju sukces nie uzasadnia twierdzenia, że jest to właściwy sposób badania SI. Definiowanie inteligencji jako „zestawu narzędzi funkcji” ma poważne słabości. Gdy jest określona w izolacji, zaimplementowana funkcja często znacznie różni się od swojej „naturalnej formy” w umyśle ludzkim. Na przykład badanie analogii bez percepcji prowadzi do zniekształconych modeli poznawczych [5]. Nawet jeśli możemy wyprodukować pożądane narzędzia, nie oznacza to, że możemy je łatwo łączyć, ponieważ różne narzędzia mogą być opracowywane przy różnych założeniach, co uniemożliwia ich łączenie. Podstawowym problemem podejścia „skrzynki z narzędziami” jest to, że bez „ogólnego obrazu” w umyśle badanie funkcji poznawczej w odizolowanej, abstrakcyjnej i często zniekształconej formie po prostu nie przyczynia się do naszego zrozumienia inteligencji. Typowy kontrargument brzmi mniej więcej tak: „Inteligencja jest bardzo złożona, więc musimy zacząć od pojedynczej funkcji, aby uczynić badanie wykonalnym”. W przypadku wielu systemów o słabych połączeniach wewnętrznych jest to często dobry wybór, ale w przypadku systemu takiego jak umysł, którego złożoność wynika bezpośrednio z jego splątanych wewnętrznych interakcji, sytuacja może być dokładnie odwrotna. Gdy tak zwane „funkcje” są w rzeczywistości zjawiskami wytwarzanymi przez złożony, ale zunifikowany mechanizm, odtworzenie ich wszystkich razem (poprzez powielenie mechanizmu) jest prostsze niż odtworzenie tylko jednej z nich.

## **Aby opracować nowe zasady**

Według tego typu opinii, to co odróżnia inteligentne systemy od nieinteligentnych systemów to ich postulaty, stosowne środowiska i podstawowe zasady przetwarzania informacji. Robocza definicja inteligencji wprowadzona wcześniej należy do tej kategorii. Jako system dostosowujący się do swojego otoczenia przy niewystarczającej wiedzy i zasobach, inteligentny system powinien mieć wiele funkcji poznawczych, ale lepiej myśleć o nich jako o zjawiskach wyłaniających się niż jako o dobrze zdefiniowanych narzędziach używanych przez system. Ucząc się na podstawie swojego doświadczenia, system potencjalnie może nabyć zdolność rozwiązywania trudnych problemów – w rzeczywistości trudne problemy to takie, dla których rozwiązujący (człowiek lub komputer) nie ma wystarczającej wiedzy i zasobów – ale nie ma takiej wbudowanej zdolności, a zatem bez odpowiedniego przeszkolenia żadna zdolność nie jest gwarantowana, a nabyte zdolności mogą nawet zostać utracone. Ponieważ umysł ludzki również kieruje się powyższymi zasadami, mamy nadzieję, że taki system będzie zachowywał się podobnie do ludzi, ale podobieństwo będzie istniało na bardziej abstrakcyjnym poziomie niż konkretne zachowania. Ze względu na fundamentalną różnicę między doświadczeniem/sprzętem ludzkim a doświadczeniem/sprzętem komputerowym, nie oczekuje się, że system będzie dokładnie odtwarzał masy danych psychologicznych ani że zda test Turinga. Wreszcie, chociaż wewnętrzna struktura systemu ma pewne właściwości wspólne z opisem umysłu ludzkiego na poziomie subsymbolicznym, nie jest to próba symulacji biologicznej sieci neuronowej. Podsumowując, podejście strukturalne przyczynia się do neuronauki poprzez budowanie modeli mózgu, podejście behawioralne przyczynia się do psychologii poprzez dostarczanie wyjaśnień dotyczących zachowań ludzkich, podejście pojemnościowe przyczynia się do domen aplikacji poprzez rozwiązywanie problemów praktycznych, a podejście funkcyjne przyczynia się do informatyki poprzez tworzenie nowego oprogramowania i sprzętu do różnych zadań obliczeniowych. Chociaż wszystkie te podejścia są cenne z różnych powodów i pomocne w dążeniu do AI, moim zdaniem podejścia te nie koncentrują się na istocie inteligencji. Oczywiście, to, co zaproponowano w mojej definicji inteligencji, nie jest zupełnie nowe dla społeczności AI. Niewielu kwestionowałoby tezę, że adaptacja lub uczenie się są niezbędne dla inteligencji. Ponadto „niewystarczająca wiedza i zasoby” są przedmiotem zainteresowania wielu poddziedzin AI, takich jak wyszukiwanie heurystyczne, rozumowanie w warunkach niepewności, planowanie w czasie rzeczywistym i uczenie maszynowe. Biorąc pod uwagę tę sytuację, co jest nowego w tym podejściu? Jest to następujący zestaw zasad:

1. wyraźna i jednoznaczna definicja inteligencji jako „adaptacji przy niewystarczającej wiedzy i zasobach”;
2. dalsza definicja zwrotu „przy niewystarczającej wiedzy i zasobach” jako skończona, w czasie rzeczywistym i otwarta;
3. projektowanie wszystkich formalnych i obliczeniowych aspektów projektu, mając przede wszystkim na uwadze dwie poprzednie definicje.

### Systemy logiczne i rozumowania

Aby uczynić naszą dyskusję bardziej konkretną i owocną, zastosujmy powyższą roboczą definicję inteligencji do szczególnego typu systemu przetwarzania informacji – systemu rozumowania. System rozumowania zazwyczaj ma następujące komponenty:

1. formalny język do reprezentacji wiedzy, a także komunikacji między systemem a jego otoczeniem;
2. semantykę, która określa znaczenia słów i wartości logiczne zdań w języku;
3. zestaw reguł wnioskowania, które dopasowują pytania do wiedzy, wyciągają wnioski z obietnic itd.;

4. pamięć, która systematycznie przechowuje zarówno pytania, jak i wiedzę, i zapewnia miejsce pracy dla wnioskowania;

5. mechanizm kontrolny, który jest odpowiedzialny za wybór przesłanek i reguł wnioskowania dla każdego etapu wnioskowania.

Pierwsze trzy komponenty są zwykle określane jako logika lub logiczna część systemu rozumowania, a ostatnie dwa jako część kontrolna systemu. Zgodnie z poprzednią definicją bycie systemem rozumowania nie jest ani konieczne, ani wystarczające do bycia inteligentnym. Jednakże inteligentny system rozumowania zapewnia odpowiednie ramy do badania inteligencji z następujących powodów:

- Jest to system ogólnego przeznaczenia. Praca w takich ramach chroni nas przed przeszkadzaniem nam przez właściwości specyficzne dla danej dziedziny, a także zapobiega oszukiwaniu poprzez stosowanie sztuczek specyficznych dla danej dziedziny.
- W porównaniu z czynnościami poznawczymi, takimi jak percepcja niskiego poziomu i kontrola motoryczna, rozumowanie jest na bardziej abstrakcyjnym poziomie i jest jedną z umiejętności poznawczych, które łącznie sprawiają, że ludzie tak jakościowo różnią się od innych zwierząt.
- Ramy systemu rozumowania są wysoce elastyczne i zbędne. Zobaczmy, że możemy w nich wykonywać wiele innych czynności poznawczych, gdy koncepcja „rozumowania” zostanie odpowiednio rozszerzona.
- Większość badań nad systemami rozumowania jest przeprowadzana w ramach paradygmatu opartego na założeniach bezpośrednio przeciwstawnych do przedstawionego powyżej. Poprzez „walkę na podwórku rywala” możemy wyraźniej zobaczyć, jakie rodzaje efektów mają nowe idee.

Zanim pokażemy, jak zaprojektowano inteligentny system rozumowania, najpierw zobaczymy jego przeciwieństwo – to znaczy system rozumowania zaprojektowany przy założeniu, że jego wiedza i zasoby są wystarczające, aby odpowiedzieć na pytania zadawane przez jego otoczenie (więc nie jest potrzebna żadna adaptacja). Z definicji taki system ma następujące właściwości:

1. Nie jest potrzebna żadna nowa wiedza. Wszystko, co system musi wiedzieć, aby odpowiedzieć na pytania, jest już na samym początku, wyrażone za pomocą zestawu aksjomatów.
2. Aksjomaty są prawdziwe i pozostaną prawdziwe w tym sensie, że odpowiadają rzeczywistej sytuacji w środowisku.
3. System odpowiada na pytania, stosując zestaw formalnych reguł do aksjomatów. Reguły są solidne i kompletne (w odniesieniu do ważnych pytań), dlatego gwarantują poprawne odpowiedzi na wszystkie pytania.
4. Pamięć systemu jest tak duża, że wszystkie aksjomaty i wyniki pośrednie zawsze mogą się w niej znaleźć.
5. Istnieje algorytm, który może przeprowadzić dowolne wymagane wnioskowanie w skończonym czasie i działa tak szybko, że może spełnić wszystkie wymagania czasowe, które mogą być przypisane do pytań.

To jest typ systemu, o którym marzyli Leibniz, Boole, Hilbert i wielu innych. Zwykle określa się go jako „rozstrzygalny system aksjomatyczny” lub „system formalny”. Próba zbudowania takich systemów zdominowała badania logiki przez stulecie i wywarła silny wpływ na badania nad sztuczną inteligencją. Wielu badaczy uważa, że taki system może służyć jako model ludzkiego myślenia. Jednak jeśli inteligencję definiuje się jako „adaptację przy niewystarczającej wiedzy i zasobach”, to chcemy czegoś



przeciwego, w pewnym sensie, do systemu aksjomatycznego, chociaż jest on nadal sformalizowany lub symbolizowany w sensie technicznym. Dlatego też wybrano system rozumowania nieaksjomatycznego (NARS) jako nazwę inteligentnego systemu rozumowania, który zostanie przedstawiony w kolejnych sekcjach. Pomiędzy systemami „czysto aksjomatycznymi” i „nieaksjomatycznymi” istnieją również systemy „półaksjomatyczne”. Są one projektowane przy założeniu, że wiedza i zasoby są niewystarczające w niektórych, ale nie we wszystkich aspektach. W związku z tym konieczna jest adaptacja. Większość obecnych systemów rozumowania opracowanych dla AI należy do tej kategorii. Zgodnie z naszą roboczą definicją inteligencji, systemy czysto aksjomatyczne w ogóle nie są inteligentne, systemy nieaksjomatyczne są inteligentne, a systemy półaksjomatyczne są inteligentne w pewnych aspektach. Systemy czysto aksjomatyczne są bardzo przydatne w matematyce, gdzie celem jest idealizacja wiedzy i pytań do takiego stopnia, że można zignorować rewizję wiedzy i terminy pytań. W takich sytuacjach na pytania można odpowiedzieć tak dokładnie i niezawodnie, że procedurę można odtworzyć za pomocą maszyny Turinga. Potrzebujemy inteligencji tylko wtedy, gdy nie można użyć takiej czysto aksjomatycznej metody z powodu niewystarczającej wiedzy i zasobów. Z tego samego powodu wydajność systemu nieaksjomatycznego niekoniecznie jest lepsza niż wydajność systemu półaksjomatycznego, ale może on działać w środowiskach, w których ten ostatni nie może być użyty. Zgodnie z powyższymi definicjami inteligencja jest nadal (jak mamy nadzieję) kwestią stopnia. Nie wszystkie systemy w kategoriach „nieaksjomatycznych” i „półaksjomatycznych” są równie inteligentne. Niektóre systemy mogą być bardziej inteligentne niż inne ze względu na wyższą efektywność zasobów, wykorzystywanie wiedzy na więcej sposobów, komunikowanie się ze środowiskiem w bogatszym języku, szybszą i dokładniejszą adaptację itd. „Nieaksjomatyczny” nie oznacza „wszystko się zmienia”. W NARS nic nie jest stałe, jeśli chodzi o treść wiedzy, ale jak zobaczymy w kolejnych sekcjach, sposób, w jaki zachodzą zmiany, jest stały, zgodnie z regułami wnioskowania i strategią kontroli systemu, które pozostają stałe, gdy system działa. Ten fakt nie czyni NARS „półaksjomatycznym”, ponieważ stała część nie znajduje się na poziomie „języka obiektowego”, ale na poziomie „metajęzyka”. W pewnym sensie możemy powiedzieć, że „meta-poziom” NARS nie jest nieaksjomatyczny, ale czysto aksjomatyczny. W przypadku systemu rozumującego stała reguła wnioskowania nie jest tym samym, co aksjomat. Oczywiście, możemy pozwolić, aby „meta-poziom” NARS był również nieaksjomatyczny, a zatem dać systemowi większą elastyczność w jego adaptacji. Jednak takie podejście nie jest przyjęte w NARS na obecnym etapie z następujących powodów:

- „Całkowita samomodifikująca się” jest iluzją. Jak powiedział Hofstadter, „poniżej każdej splątanej hierarchii leży nienaruszalny poziom”. Jeśli pozwolimy NARS modyfikować jego wiedzę metapoziomu, tj. jego reguły wnioskowania i strategię kontroli, musimy dać mu (stałą) wiedzę meta-meta-poziomu, aby określić, w jaki sposób modyfikacja zachodzi. Choć ludzki umysł jest elastyczny, nie może modyfikować swojego własnego „prawa myśli”.
- Chociaż samomodifikująca się na wysokim poziomie nada systemowi większą elastyczność, niekoniecznie czyni go bardziej inteligentnym. Samomodifikująca się na poziomie meta jest często niebezpieczna i powinna być stosowana tylko wtedy, gdy nie można uzyskać tego samego efektu na poziomie obiektu. Założenie, że „im bardziej radykalne mogą być zmiany, tym bardziej inteligentny będzie system”, jest bezpodstawne. Łatwo jest pozwolić systemowi modyfikować swój własny kod źródłowy, ale trudno zrobić to dobrze.
- W przyszłości będziemy badać możliwość uczenia się na poziomie meta w NARS, ale nie będziemy tego próbować robić, dopóki uczenie się na poziomie obiektów nie dojrzeje. Próbowanie wszystkiego w tym samym czasie nie jest po prostu dobrym podejściem inżynierskim i nie czyni to NARS mniej nieaksjomatycznym, zgodnie z powyższą definicją.

Wiele argumentów przedstawionych wcześniej przeciwko logicznej AI, symbolicznej AI lub AI jako całości, jest w rzeczywistości przeciwko bardziej konkretnemu celowi: systemom czysto aksjomatycznym. Te argumenty są potężne, ponieważ ujawniają, że wiele aspektów inteligencji nie może być wytworzonych przez system czysto aksjomatyczny (choć autorzy ci nie używają tego terminu), ale niektóre z nich są mylące, ponieważ używają takiego systemu jako prototypu badań nad AI. Pracując nad systemem rozumowania, z jego formalnym językiem i regułami wnioskowania, niekoniecznie wiążemy się zobowiązaniami akceptowanymi przez tradycyjne paradygmaty „logicznej AI”. Jak zobaczymy w dalszej części, NARS dzieli bardziej filozoficzne opinie z ruchem subsymbolicznym lub koneksjonistycznym. Jaki jest związek sztucznej inteligencji i informatyki? Jakie jest miejsce AI w całym przedsięwzięciu naukowym? Tradycyjnie AI jest określane jako gałąź informatyki. Zgodnie z naszymi poprzednimi definicjami, sztuczną inteligencję można wdrożyć za pomocą narzędzi dostarczanych przez informatykę, ale z teoretycznego punktu widzenia przyjmują one przeciwne założenia: informatyka koncentruje się na systemach czysto aksjomatycznych, podczas gdy sztuczna inteligencja koncentruje się na systemach nieaksjomatycznych. Podstawowe założenia informatyki można znaleźć w logice matematycznej (zwłaszcza logice predykatów pierwszego rzędu) i teorii obliczalności (zwłaszcza maszynie Turinga). Teorie te przyjmują wystarczalność wiedzy i zasobów jako postulaty domyślne, dlatego adaptacja, prawdopodobne wnioskowanie i wstępne rozwiązania problemów nie są ani konieczne, ani możliwe. Podobne założenia są często akceptowane przez badaczy sztucznej inteligencji z następującym uzasadnieniem: „Wiemy, że ludzki umysł zwykle pracuje z niewystarczającą wiedzą i zasobami, ale jeśli chcesz utworzyć model formalny, a następnie system komputerowy, musisz w jakiś sposób zidealizować sytuację”. Prawdą jest, że każdy model formalny jest idealizacją, podobnie jak NARS. Problem polega na tym, co pominąć, a co zachować w idealizacji. W obecnej implementacji NARS wiele czynników, które powinny wpływać na rozumowanie, jest ignorowanych, ale niedobór wiedzy i zasobów jest ściśle zakładany przez cały czas. Dlaczego? Ponieważ jest to definitywna cecha inteligencji, więc gdyby została utracona przez „idealizację”, wynikowe badanie dotyczyłoby czegoś innego.

### **Składniki NARS**

Nieaksjomatyczny system rozumowania (NARS) został zaprojektowany jako inteligentny system rozumowania, zgodnie z roboczą definicją inteligencji wprowadzoną wcześniej. W dalszej części zobaczymy, w jaki sposób główne składniki NARS (jego formalny język, semantyka, reguły wnioskowania, pamięć i mechanizm kontroli) są określone lub silnie sugerowane przez definicję i w jaki sposób różnią się od składników systemu aksjomatycznego. Ponieważ ten rozdział koncentruje się na filozoficznych i metodologicznych podstawach projektu NARS, formalne opisy i szczegółowe dyskusje na temat składników pozostawiono innym pracom.

### **Semantyka oparta na doświadczeniu**

Systemy rozumowania aksjomatycznego (i większość systemów semiaksjomatycznych) używają „semantyki teoretyczno-modelowej”. Mówiąc nieformalnie, model jest opisem domeny, z określonymi relacjami między obiektami. W przypadku systemu rozumowania działającego w domenie „interpretacja” mapuje terminy w systemie na obiekty w modelu, a predykaty w systemach na relacje w modelu. Dla danego terminu jego znaczeniem jest jego obraz w modelu pod interpretacją. Dla danego stwierdzenia jego wartość logiczna zależy od tego, czy odpowiada faktowi w modelu. Przy takiej semantyce system rozumowania otrzymuje stałe „odniesienie”, model, zgodnie z którym określana jest prawda i znaczenie w systemie. Chociaż semantyka teoretyczno-modelowa występuje w różnych formach i ma odmiany, ten „ogólny obraz” pozostaje niezmienny. Ten rodzaj semantyki nie nadaje się dla NARS. Jako system adaptacyjny z niewystarczającą wiedzą i zasobami, system nie może oceniać prawdziwości swojej wiedzy w odniesieniu do statycznego, spójnego i kompletnego

modelu. Zamiast tego prawda i znaczenie muszą być oparte na doświadczeniu systemu [40]. Chociaż część doświadczenia jest również opisem otoczenia systemu, zasadniczo różni się od modelu, ponieważ doświadczenie zmienia się w czasie, nigdy nie jest kompletne i często jest niespójne. Ponadto doświadczenie jest bezpośrednio dostępne dla systemu, podczas gdy model jest często „w oczach obserwatora”. Zgodnie z semantyką opartą na doświadczeniu, wartość prawdy staje się funkcją ilości dostępnych dowodów, dlatego nieuchronnie staje się zmienna i subiektywna, choć nie arbitralna. W takim systemie żadna wiedza nie jest „prawdziwa” w tym sensie, że gwarantuje się jej potwierdzenie przez przyszłe doświadczenie. Zamiast tego wartość prawdy stwierdzenia wskazuje stopień, w jakim stwierdzenie jest potwierdzone przez przeszłe doświadczenie. System wykorzysta taką wiedzę do przewidywania przyszłości, ponieważ jest to dokładnie to, co oznacza „adaptacyjny”, a zatem „inteligentny”. W ten sposób „prawda” ma zupełnie inne (choć ściśle powiązane) znaczenia w systemach nieaksjomatycznych i aksjomatycznych. Podobnie znaczenie terminu, czyli to, co odróżnia termin od innych terminów w systemie, jest określone przez jego relacje z innymi terminami, zgodnie z doświadczeniem systemu, a nie przez interpretację, która mapuje go na obiekt w modelu. Przy niewystarczających zasobach wartość logiczna każdego stwierdzenia i znaczenie każdego terminu w NARS są zwykle oparte na części doświadczenia. W rezultacie, nawet bez nowego doświadczenia, aktywność wnioskowania systemu zmieni wartości logiczne i znaczenia, biorąc pod uwagę wcześniej dostępne, ale zignorowane doświadczenie. Wręcz przeciwnie, zgodnie z semantyką teorii modeli, wewnętrzne działania systemu nie mają wpływu na wartość logiczną i znaczenie języka, którego używa. „Bez interpretacji system nie ma dostępu do semantyki języka formalnego, którego używa” jest centralnym argumentem w eksperymencie myślowym Searle’a „Chinese room” przeciwko silnej sztucznej inteligencji. Jego argument jest ważny dla semantyki teoretyczno-modelowej, ale nie dla semantyki opartej na doświadczeniu. Dla inteligentnego systemu rozumowania bardziej odpowiednia jest ta druga.

### **Oświadczenie o dziedziczeniu**

Jak omówiono powyżej, „adaptacja przy niewystarczającej wiedzy i zasobach” wymaga semantyki opartej na doświadczeniu, która z kolei wymaga formalnego języka reprezentacji wiedzy, w którym dowody mogą być naturalnie definiowane i mierzone. W przypadku systemu rozumowania nieaksjomatycznego oczywiste jest, że binarna wartość prawdy nie jest wystarczająca. Mając za jedyne wskazówki przeszłe doświadczenie, system nie tylko musi wiedzieć, czy istnieje kontrprzykład (negatywny dowód), ale także musi znać jego ilość w odniesieniu do ilości pozytywnych dowodów. Aby mieć niezależną od domeny metodę porównywania konkurencyjnych odpowiedzi, konieczna staje się wartość prawdy liczbowej lub pomiar niepewności dla NARS, która ilościowo rejestruje związek między stwierdzeniem a dostępnymi dowodami. Ponadto należy również rozróżnić „pozytywne dowody” i „nieistotne rzeczy”. Intuicyjnie rzecz biorąc, najprostszym przypadkiem zdefiniowania dowodu jest ogólne stwierdzenie dotyczące wielu przypadków, podczas gdy niektóre z nich są potwierdzone przez przeszłe doświadczenie (pozytywne dowody), a niektóre inne są podważane przez przeszłe doświadczenie (negatywne dowody). Niestety, najpopularniejszy formalny język reprezentacji wiedzy, rachunek predykatów pierwszego rzędu, nie może być łatwo używany w ten sposób. W tym języku „ogólne stwierdzenie”, takie jak „Kruki są czarne”, jest reprezentowane jako „uniwersalne twierdzenie”, takie jak „ $(\forall x)(\text{Kruk}(x) \rightarrow \text{Czarny}(x))$ ”. W pierwotnej formie rachunku predykatów pierwszego rzędu nie ma takiego pojęcia jak „dowód”, a twierdzenie jest albo prawdziwe, albo fałszywe, w zależności od tego, czy w domenie istnieje taki obiekt  $x$ , który sprawia, że  $\text{Raven}(x)$  jest prawdziwy, a  $\text{Black}(x)$  fałszywy. Naturalne jest definiowanie stałych, które czynią twierdzenie prawdziwym, jako jego pozytywnego dowodu, a stałych, które czynią je fałszywym, jako jego negatywnego dowodu. Jednak takie naiwne rozwiązanie ma poważne problemy:

- Tylko istnienie negatywnego dowodu przyczynia się do wartości logicznej uniwersalnego twierdzenia, podczas gdy to, czy istnieje „pozytywny dowód”, nie ma znaczenia. To jest początek teorii obalenia Poppera.

- Każda stała jest albo pozytywnym, albo negatywnym dowodem, i nic nie jest nieistotne. Jest to związane z paradoksem konformacji Hempela.

Chociaż dowody trudno zdefiniować w rachunku predykatów, łatwo to zrobić w odpowiednio zaprojektowanej logice kategoriowej. Logiki kategoriowe, lub logiki terminów, to kolejna rodzina logiki formalnej, której przykładem jest sylogizm Arystotelesa. Głównymi cechami formalnymi, które odróżniają ją od logiki predykatów, są użycie stwierżeń podmiotowo-predykatowych i reguł wnioskowania sylogistycznego. Zaczniemy od pierwszej cechy. NARS używa języka kategoriowego opartego na relacji dziedziczenia „ $\rightarrow$ ”. Relacja ta w swojej idealnej formie jest zwrotną i przechodnią relacją binarną zdefiniowaną na terminach, gdzie termin można uważać za nazwę pojęcia. Na przykład „ $raven \rightarrow bird$ ” jest stwierdzeniem dziedziczenia z „raven” jako terminem podmiotowym i „bird” jako terminem predykatu. Intuicyjnie mówi, że podmiot jest specjalizacją predykatu, a predykat jest uogólnieniem podmiotu. Stwierdzenie to mniej więcej odpowiada angielskiemu zdaniu „Raven is a kind of bird”. Na podstawie relacji dziedziczenia, rozszerzenie i intencja terminu są definiowane jako zbiór jego specjalizacji i uogólnień, odpowiednio. Oznacza to, że dla danego terminu T, jego rozszerzenie T E jest zbiorem  $\{x \mid x \rightarrow T\}$ , a jego intencja T I jest zbiorem  $\{x \mid T \rightarrow x\}$ . Biorąc pod uwagę zwrotność i przechodniość relacji dziedziczenia, można udowodnić, że dla dowolnych terminów S i P, „ $S \rightarrow P$ ” jest prawdziwe wtedy i tylko wtedy, gdy SE jest zawarte w PE, a PI jest zawarte w SI. Innymi słowy, „Istnieje relacja dziedziczenia z S do P” jest równoważne z „P dziedziczy rozszerzenie S, a S dziedziczy intencję P”. Rozważając „niedoskonałe” stwierdzenia dziedziczenia, powyższe twierdzenie naturalnie daje nam definicję (pozytywnego i negatywnego) dowodu. Dla danego stwierdzenia „ $S \rightarrow P$ ”, jeśli wyraz M jest zarówno w SE, jak i PE, lub zarówno w PI, jak i SI, to jest to pozytywny dowód na to stwierdzenie, ponieważ jeśli chodzi o M, proponowane dziedziczenie jest prawdziwe; jeśli M jest w SE, ale nie w PE, lub w PI, ale nie w SI, to jest to negatywny dowód na to stwierdzenie, ponieważ jeśli chodzi o M, proponowane dziedziczenie jest fałszywe; jeśli M nie jest ani w SE, ani w PI, to nie jest to dowód na to stwierdzenie i nie ma znaczenia, czy jest również w PE lub SI. Użyjmy  $w+$ ,  $w-$  i  $w$  dla ilości dowodów pozytywnych, negatywnych i całkowitych, odpowiednio, wtedy mamy  $w+ = |SE \cap PE| + |PI \cap SI|$ ,  $w- = |SE - PE| + |PI - SI|$ ,  $w = w++w- = |SE| + |PI|$ . Na koniec definiujemy wartość logiczną stwierdzenia jako parę liczb  $\langle f, c \rangle$ . Tutaj  $f$  nazywane jest częstością stwierdzenia, a  $f = w+/w$ . Drugi składnik  $c$  nazywany jest ufnością stwierdzenia, a  $c = w/(w+k)$ , gdzie  $k$  jest parametrem systemu, którego wartością domyślną jest 1. Bardziej szczegółową dyskusję można znaleźć w [43]. Teraz mamy techniczne podstawy semantyki opartej na doświadczeniu: Jeśli doświadczenie systemu jest zbiorem stwierżeń dziedziczenia zdefiniowanych powyżej, to dla dowolnego terminu T możemy określić jego znaczenie, które jest jego rozszerzeniem i intencją (zgodnie z doświadczeniem), a dla dowolnego stwierdzenia dziedziczenia „ $S \rightarrow P$ ” możemy określić jego dowód pozytywny i dowód negatywny (poprzez porównanie znaczenia dwóch terminów), a następnie obliczyć jego wartość logiczną zgodnie z powyższą definicją. Oczywiście, rzeczywiste doświadczenie NARS nie jest zbiorem binarnych stwierżeń dziedziczenia, ani system nie określa wartości logicznej stwierdzenia w powyższy sposób. Rzeczywiste doświadczenie NARS to strumień stwierżeń, których wartości logiczne są reprezentowane przez pary  $\langle f, c \rangle$ . W ramach systemu nowe stwierdzenia są wyprowadzane przez reguły wnioskowania, przy czym funkcje wartości logicznej obliczają wartości logiczne wniosków z tych z przesłanek. Celem powyższych definicji jest zdefiniowanie wartości logicznej w zidealizowanej sytuacji i zapewnienie podstaw dla funkcji wartości logicznej.

## Język kategoriowy

Na podstawie relacji dziedziczenia wprowadzonej powyżej, NARS używa potężnego „języka kategoriowego”, uzyskanego przez rozszerzenie powyższego języka podstawowego w różnych kierunkach:

Pochodne relacje dziedziczenia: Oprócz relacji dziedziczenia zdefiniowanej wcześniej, NARS obejmuje również kilka jej wariantów. Na przykład,

- relacja podobieństwa  $\leftrightarrow$  jest dziedziczeniem symetrycznym;
- relacja instancji  $\circ \rightarrow$  jest relacją dziedziczenia, w której termin podmiotowy jest traktowany jako atomowa instancja terminu predykatu;
- relacja własności  $\rightarrow \circ$  jest relacją dziedziczenia, w której termin predykatu jest traktowany jako pierwotna własność terminu podmiotowego.

Terminy złożone: W instrukcjach dziedziczenia terminy (podmiotu i predykatu) mogą być nie tylko prostymi nazwami (jak w powyższych przykładach), ale także mogą być terminami złożonymi utworzonymi przez inne terminy z operatorem logicznym. Na przykład, jeśli A i B są terminami, mamy

- ich przecięcie ekstensjonalne  $(A \cap B)$  jest terminem złożonym, zdefiniowanym przez  $(A \cap B)E = (AE \cap BE)$  i  $(A \cap B)I = (AI \cup BI)$ .
- ich przecięcie intencjonalne  $(A \cup B)$  jest terminem złożonym, zdefiniowanym przez

$$(A \cup B)E = (AE \cup BE) \text{ i } (A \cup B)I = (AI \cap BI);$$

W przypadku terminów złożonych siła wyrazu języka jest znacznie rozszerzona.

Zwykła relacja: W NARS tylko relacja dziedziczenia i jej warianty są definiowane jako stałe logiczne, które są bezpośrednio rozpoznawane przez reguły wnioskowania. Wszystkie inne relacje są przekształcane w relacje dziedziczenia z terminami złożonymi. Na przykład, dowolna relacja R pomiędzy trzema terminami A, B i C jest zwykle zapisywana jako  $R(A,B,C)$ , co można równoważnie zapisać jako jedno z następujących oświadczeń dziedziczenia (tj. mają one to samo znaczenie i wartość logiczną):

- „ $(A,B,C) \rightarrow R$ ”, gdzie termin podmiotowy jest złożonym  $(A,B,C)$ , uporządkowaną krotką. To oświadczenie mówi „Relacja pomiędzy A, B, C (w tej kolejności) jest wystąpieniem relacji R”.
- „ $A \rightarrow R(*,B, C)$ ”, gdzie termin predykatu jest złożonym  $R(*,B, C)$  z „kartą dziką”, \* . To oświadczenie mówi „A jest takim x, które spełnia  $R(x,B,C)$ .”
- „ $B \rightarrow R(A, *, C)$ ”. Podobnie „B jest takim x, które spełnia  $R(A, x,C)$ .”
- „ $C \rightarrow R(A,B, *)$ ”. Ponownie, „C jest takim x, które spełnia  $R(A,B, x)$ .”

Termin wyższego rzędu: W NARS stwierdzenie może być używane jako termin, który jest nazywany terminem „wyższego rzędu”. Na przykład „Ptak jest rodzajem zwierzęcia” jest reprezentowane przez stwierdzenie „ $\text{ptak} \rightarrow \text{zwierzę}$ ”, a „Tom wie, że ptak jest rodzajem zwierzęcia” jest reprezentowane przez stwierdzenie „ $(\text{ptak} \rightarrow \text{zwierzę}) \circ \rightarrow \text{wiem}(\text{Tom}, *)$ ”, gdzie termin podmiotowy jest stwierdzeniem. Złożone terminy wyższego rzędu są również definiowane: jeśli A i B są terminami wyższego rzędu, to ich negacja  $(\neg A \text{ i } \neg B)$ , alternatywa  $(A \vee B)$  i koniunkcja  $(A \wedge B)$ . Relacja wyższego rzędu: Relacje wyższego rzędu to te, których termin podmiotowy i termin orzeka są terminami wyższego rzędu. W NARS istnieją dwie zdefiniowane jako stałe logiczne:

- implikacja, „ $\Rightarrow$ ”, która intuicyjnie odpowiada „jeśli-to”;

- równoważność, „ $\Leftrightarrow$ ”, która intuicyjnie odpowiada „jeśli i tylko jeśli”.

Zdania niedeklaratywne: Oprócz różnych typów stwierdzeń wprowadzonych powyżej, które reprezentują deklaratywną wiedzę systemu, formalny język NARS używa podobnych formatów do reprezentowania zdań niedeklaratywnych:

- pytanie jest albo stwierdzeniem, którego wartość logiczna musi zostać oceniona (pytania „tak/nie”), albo stwierdzeniem zawierającym zmienne do utworzenia (pytania „co”);
- cel jest stwierdzeniem, którego prawdziwość musi zostać ustalona przez system poprzez wykonanie odpowiednich operacji.

Dla każdego typu stwierdzeń jego wartość logiczna jest definiowana podobnie do tego, jak definiujemy wartość logiczną stwierdzenia dziedziczenia. Przy powyższych strukturach siła ekspresji języka jest mniej więcej taka sama jak typowego języka naturalnego (takiego jak angielski lub chiński). Nie ma odwzorowania jeden do jednego między zdaniami w tym języku a zdaniami w rachunku predykatów pierwszego rzędu, chociaż przybliżone odwzorowanie jest możliwe dla wielu zdań. Podczas gdy rachunek predykatów pierwszego rzędu może nadal lepiej reprezentować wiedzę matematyczną, ten nowy język będzie lepszy do reprezentowania wiedzy empirycznej (które zostaną omówione poniżej).

### Reguły wnioskowania sylogistycznego

Z powodu niewystarczającej wiedzy system musi przeprowadzać wnioskowanie niededukcyjne, takie jak indukcja, abdukcja i analogia, aby rozszerzyć przeszłe doświadczenie na nowe sytuacje. W tym kontekście dedukcja staje się również omylna w tym sensie, że jej wniosek może zostać zmieniony przez nową wiedzę, nawet jeśli przesłanki pozostają niepodważalne. Zgodnie z semantyką ugruntowaną na doświadczeniu definicja ważności reguł wnioskowania ulega zmianie. Zamiast generować nieomylnie wnioski, prawidłowa reguła powinna generować wnioski, których wartości logiczne są oceniane na podstawie (i tylko na podstawie) dowodów dostarczonych przez przesłanki. Jak wspomniano wcześniej, główną cechą odróżniającą logikę terminów od logiki predykatów/propozycjonalnej jest stosowanie reguł wnioskowania sylogistycznego, z których każda przyjmuje parę przesłanek, które mają wspólny termin. W przypadku wnioskowania między stwierdzeniami dziedziczenia istnieją trzy możliwe kombinacje, jeśli dwie przesłanki mają dokładnie jeden termin:

deduction	induction	abduction
$M \rightarrow P \langle f_1, c_1 \rangle$	$M \rightarrow P \langle f_1, c_1 \rangle$	$P \rightarrow M \langle f_1, c_1 \rangle$
$S \rightarrow M \langle f_2, c_2 \rangle$	$M \rightarrow S \langle f_2, c_2 \rangle$	$S \rightarrow M \langle f_2, c_2 \rangle$
$S \rightarrow P \langle f, c \rangle$	$S \rightarrow P \langle f, c \rangle$	$S \rightarrow P \langle f, c \rangle$

Każda reguła wnioskowania ma własną funkcję wartości logicznej do obliczania wartości logicznej wniosku zgodnie z wartościami przesłanek. W NARS funkcje te są zaprojektowane w następujący sposób:

1. Traktuj wszystkie istotne zmienne jako zmienne binarne przyjmujące wartości 0 lub 1 i określ, jakie wartości wniosek powinien mieć dla każdej kombinacji przesłanek, zgodnie z semantyką.
2. Przedstaw wartości logiczne wniosku uzyskanego powyżej jako funkcje logiczne tych z przesłanek.
3. Rozszerz operatory logiczne na funkcje liczb rzeczywistych zdefiniowane na  $[0, 1]$  w następujący sposób:

$$\begin{aligned} \text{not}(x) &= 1 - x \\ \text{and}(x_1, \dots, x_n) &= x_1 * \dots * x_n \\ \text{or}(x_1, \dots, x_n) &= 1 - (1 - x_1) * \dots * (1 - x_n) \end{aligned}$$

4. Użyj rozszerzonych operatorów, plus relacji między wartością prawdy a ilością dowodów, aby przepisać powyższe funkcje. Wynik jest następujący:

$$\begin{array}{lll} \text{deduction} & \text{induction} & \text{abduction} \\ f = f_1 f_2 & f = f_1 & f = f_2 \\ c = c_1 c_2 f_1 f_2 & c = f_2 c_1 c_2 / (f_2 c_1 c_2 + 1) & c = f_1 c_1 c_2 / (f_1 c_1 c_2 + 1) \end{array}$$

Jeżeli dwie przesłanki zawierają to samo stwierdzenie, ale pochodzą z różnych części doświadczenia, stosuje się regułę rewizji, aby połączyć je w jeden podsumowujący wniosek:

$$\begin{array}{l} \text{revision} \\ S \rightarrow P \langle f_1, c_1 \rangle \\ S \rightarrow P \langle f_2, c_2 \rangle \\ \hline S \rightarrow P \langle f, c \rangle \end{array}$$

Ponieważ w rewizji dowodem na wniosek jest suma dowodów w przesłankach, funkcja wartości logicznej jest

$$\begin{aligned} f &= \frac{f_1 c_1 / (1 - c_1) + f_2 c_2 / (1 - c_2)}{c_1 / (1 - c_1) + c_2 / (1 - c_2)} \\ c &= \frac{c_1 / (1 - c_1) + c_2 / (1 - c_2)}{c_1 / (1 - c_1) + c_2 / (1 - c_2) + 1} \end{aligned}$$

Oprócz powyższych czterech podstawowych reguł wnioskowania, w NARS istnieją reguły wnioskowania dla wariantów dziedziczenia, a także dla tworzenia i przekształcania różnych terminów złożonych. Funkcje wartości logicznych dla tych reguł są podobnie określone.

Oprócz powyższych reguł wnioskowania do przodu, dzięki którym nowa wiedza jest wyprowadzana z istniejącej wiedzy, NARS ma również reguły wnioskowania wstecznego, dzięki którym część wiedzy jest stosowana do pytania lub celu. Jeśli wiedza przypadkowo dostarcza odpowiedzi na pytanie lub operację w celu realizacji celu, jest akceptowana jako rozwiązanie próbne, w przeciwnym razie może zostać wygenerowane pytanie pochodne lub cel, którego rozwiązanie, połączone z wiedzą, dostarczy rozwiązania pierwotnego pytania lub celu. Zdefiniowane w ten sposób, dla każdej reguły do przodu istnieje odpowiadająca jej reguła wsteczna. Lub, koncepcyjnie, możemy postrzegać je jako dwa sposoby wykorzystania tej samej reguły.

### Kontrolowana współbieżność w pamięci dynamicznej

Jako otwarty system działający w czasie rzeczywistym, NARS cały czas akceptuje nowe zadania. Nowe zadanie może być częścią wiedzy do przyswojenia, pytaniem do odpowiedzi lub celem do osiągnięcia. Nowe zadanie może pochodzić od użytkownika lub innego systemu komputerowego. Ponieważ w NARS żadna wiedza nie jest absolutnie prawdziwa, system będzie próbował wykorzystać jak najwięcej wiedzy do przetworzenia zadania, aby zapewnić lepsze (bardziej pewne) rozwiązanie. Z drugiej strony, z powodu niewystarczających zasobów, system nie może wykorzystać całej istotnej wiedzy do każdego zadania. Ponieważ nowe zadania pojawiają się od czasu do czasu, a system stale generuje zadania pochodne, w dowolnym momencie system zazwyczaj ma dużą liczbę zadań do przetworzenia. W tej

sytuacji zbyt sztywne jest ustalenie statycznego standardu dla satysfakcjonującego rozwiązania [35], ponieważ bez względu na to, jak ostrożnie zostanie ustalony standard, czasami będzie on zbyt wysoki, a czasami zbyt niski, biorąc pod uwagę ciągle zmieniające się zapotrzebowanie na zasoby istniejących zadań. NARS próbuje znaleźć najlepsze rozwiązanie biorąc pod uwagę aktualną wiedzę i ograniczenia zasobów [40] — podobnie jak robi to „algorytm anytime” [6]. „Torba” to struktura danych specjalnie zaprojektowana w NARS do alokacji zasobów. Torba może zawierać pewien typ elementów o stałej pojemności i utrzymuje priorytetowy rozkład między elementami. Istnieją trzy główne operacje zdefiniowane w torbie:

- Umieść element w torbie, a jeśli torba jest już pełna, usuń element o najniższym priorytecie.
- Wyjmij element z torby według klucza (tj. jego unikalnego identyfikatora).
- Wyjmij element z torby według priorytetu, czyli prawdopodobieństwo, że element zostanie wybrany, jest proporcjonalne do jego wartości priorytetu.

Każda z operacji zajmuje stały czas, niezależnie od liczby elementów w torbie. NARS organizuje wiedzę i zadania w koncepcje. W systemie termin  $T$  ma odpowiadające mu pojęcie  $CT$ , które zawiera całą wiedzę i zadania, w których  $T$  jest terminem podmiotowym lub terminem predykatu. Na przykład wiedza „ptak  $\rightarrow$  zwierzę  $\langle 1, 0,9 \rangle$ ” jest przechowywana w pojęciu  $Cbird$  i pojęciu  $Canimal$ . W ten sposób pamięć NARS można postrzegać mniej więcej jako worek pojęć, a każde pojęcie jest nazwane terminem (prostym lub złożonym) i zawiera worek wiedzy i worek zadań, wszystkie z nich bezpośrednio dotyczą terminu. NARS działa, wielokrotnie wykonując następujący cykl roboczy:

1. Weź pojęcie z pamięci według priorytetu.
2. Weź zadanie z worka zadań pojęcia według priorytetu.
3. Weź część wiedzy z worka wiedzy pojęcia według priorytetu.
4. Zgodnie z połączeniem zadania i wiedzy wywołaj obowiązujące reguły wnioskowania na nich, aby wyprowadzić nowe zadania i nową wiedzę — w logice terminów każdy krok wnioskowania zachodzi w ramach pojęcia.
5. Dostosuj priorytet zaangażowanego zadania, wiedzy i pojęcia zgodnie z tym, jak zachowują się w tym kroku wnioskowania, a następnie umieść je z powrotem w odpowiednich workach. 6. Umieść nowe (wprowadzone lub pochodne) zadania i wiedzę w odpowiednich workach. Jeśli pewna nowa wiedza dostarcza najlepszego rozwiązania do tej pory dla zadania przypisanego przez użytkownika, zgłoś rozwiązanie.

Wartość priorytetu każdego elementu odzwierciedla ilość zasobów, jaką system planuje na niego przeznaczyć w najbliższej przyszłości. Ma dwa czynniki:

Czynnik długoterminowy. System nadaje wyższy priorytet ważniejszym elementom, ocenianym na podstawie doświadczeń z przeszłości. Początkowo użytkownik może przypisać wartości priorytetów zadaniom wejściowym, aby wskazać ich względne znaczenie, co z kolei określi wartość priorytetu pojęć i wiedzy z nich wygenerowanej. Po każdym kroku wnioskowania wartości priorytetów zaangażowanych elementów są dostosowywane. Na przykład, jeśli część wiedzy zapewnia najlepsze dotychczasowe rozwiązanie zadania, wartość priorytetu wiedzy jest zwiększana (aby była częściej wykorzystywana w przyszłości), a wartość priorytetu zadania jest zmniejszana (aby w przyszłości poświęcono na nią mniej czasu).



Czynnik krótkoterminowy. System nadaje wyższy priorytet bardziej istotnym elementom, ocenianym na podstawie bieżącego kontekstu. Gdy do systemu dodawane jest nowe zadanie, bezpośrednio powiązane pojęcia są aktywowane, tj. ich wartości priorytetu są zwiększane. Z drugiej strony wartości priorytetu zanikają z czasem, więc jeśli pojęcie nie było istotne przez jakiś czas, staje się mniej aktywne.

W ten sposób NARS przetwarza wiele zadań równoległe, ale z różną prędkością. Ten mechanizm kontroli „kontrolowanej współbieżności” jest podobny do strategii „równoległego skanowania tarasowego” Hofstadtera [14]. Ponadto sposób przetwarzania zadania zależy od dostępnej wiedzy i priorytetowego rozkładu pojęć, zadań i wiedzy. Ponieważ czynniki te zmieniają się nieustannie, rozwiązanie, jakie otrzymuje zadanie, jest zależne od kontekstu.

### **Właściwości NARS**

Jako projekt ukierunkowany na sztuczną inteligencję ogólnego przeznaczenia, NARS zajmuje się wieloma zagadnieniami w dziedzinie AI i nauk kognitywnych. Chociaż jest podobny do wielu innych podejść, projekt jako całość jest wyjątkowy pod względem podstaw teoretycznych i głównych komponentów technicznych. Zaprojektowany jak powyżej, NARS wykazuje wiele właściwości, które czynią go bardziej podobnym do ludzkiego rozumowania niż inne systemy AI.

### **Rozsądne rozwiązania**

Przy niewystarczającej wiedzy i zasobach NARS nie może zagwarantować, że wszystkie rozwiązania, które generuje dla zadań, są poprawne w tym sensie, że nie zostaną zakwestionowane przez przyszłe doświadczenia systemu. Nie może również zagwarantować, że rozwiązania są optymalne, biorąc pod uwagę całą wiedzę, jaką system posiada w danym momencie. Jednak rozwiązania są rozsądne w tym sensie, że są najlepszym podsumowaniem przeszłych doświadczeń, biorąc pod uwagę obecne zasoby. Jest to podobne do „racjonalności typu II” Gooda [10]. NARS często popełnia „rozsądne błędy”, które są spowodowane niedoborem wiedzy i zasobów. Są one rozsądne i nieuniknione, biorąc pod uwagę warunki pracy systemu, i nie są spowodowane błędami w projektowaniu lub funkcji systemu. Konwencjonalny algorytm zapewnia pojedyncze rozwiązanie dla każdego problemu, a następnie przestaje pracować nad problemem. Wręcz przeciwnie, NARS może nie zapewnić żadnego, jednego lub więcej niż jednego rozwiązania dla zadania — zgłasza każde najlepsze znalezione rozwiązanie, a następnie szuka lepszego (jeśli zasoby są nadal dostępne). Oczywiście, w końcu system zakończy przetwarzanie zadania, ale powodem nie jest ani znalezienie satysfakcjonującego rozwiązania, ani upływ terminu, ale przegrana zadania w rywalizacji o zasoby. Podobnie jak procedury prób i błędów, NARS może „zmienić zdanie”. Ponieważ wartości prawdy są ustalane na podstawie doświadczenia, późniejsze rozwiązanie jest oceniane jako „lepsze” niż poprzednie, ponieważ opiera się na większej liczbie dowodów, chociaż niekoniecznie jest „bliższe obiektywnemu faktowi”. Gdy rozwiązanie zostanie znalezione, zazwyczaj nie ma sposobu, aby zdecydować, czy jest to ostatecznie, jakie system może uzyskać. W NARS nie ma „ostatecznego rozwiązania”, którego nie można zaktualizować za pomocą nowej wiedzy i/lub dalszych rozważań, ponieważ wszystkie rozwiązania opierają się na częściowym doświadczeniu systemu. Ta samorewidowalna cecha sprawia, że NARS jest bardziej ogólnym modelem niż różne logiki niemonotoniczne, w których przetwarzane są tylko instrukcje binarne, a aktualizowane mogą być tylko wnioski wyprowadzone z domyślnych reguł, ale same domyślne reguły nie są objęte doświadczeniem systemu.

### **Zunifikowane przetwarzanie niepewności**

Jak opisano wcześniej, w NARS istnieją różne rodzaje niepewności w koncepcjach, stwierdzeniach, regułach wnioskowania i procesach wnioskowania. NARS ma zunifikowany podsystem pomiaru i obliczeń niepewności. To podejście różni się od innych proponowanych teorii niepewności semantyką

opartą na doświadczeniu. Zgodnie z nią wszelka niepewność wynika z niedoboru wiedzy i zasobów. W rezultacie ocena niepewności jest zmienna i zależna od kontekstu. Z naszej poprzedniej definicji wartości prawdy łatwo rozpoznać jej związek z teorią prawdopodobieństwa. Przy pewnej interpretacji pomiar częstotliwości jest podobny do prawdopodobieństwa, a pomiar pewności jest związany z rozmiarem przestrzeni próby. Jeśli tak jest, dlaczego nie użyć bezpośrednio teorii prawdopodobieństwa do radzenia sobie z niepewnością? Zobaczmy konkretny przypadek. Reguła dedukcji przyjmuje „ $M \rightarrow P \langle f1, c1 \rangle$ ” i „ $S \rightarrow M \langle f2, c2 \rangle$ ” jako przesłanki i wyprowadza „ $S \rightarrow P \langle f, c \rangle$ ” jako wniosek. Bezpośrednim sposobem zastosowania teorii prawdopodobieństwa byłoby potraktowanie każdego członu jako zbioru, a następnie przekształcenie reguły w taką, która oblicza prawdopodobieństwo warunkowe  $Pr(P|S)$  z  $Pr(P|M)$  i  $Pr(M|S)$  plus dodatkowe założenia dotyczące funkcji rozkładu prawdopodobieństwa  $Pr()$ . Podobnie szacowany byłby rozmiar próby wniosku, co daje wartość ufności. Takiego podejścia nie można zastosować w NARS z kilku powodów:

- W przypadku relacji dziedziczenia dowód jest definiowany zarówno ekstensywnie, jak i intencjonalnie, więc częstość „ $S \rightarrow P$ ” nie może być traktowana jako  $Pr(P|S)$ , ponieważ ten ostatni jest czysto ekstensjonalny.
- Każde stwierdzenie ma własną przestrzeń dowodową, zdefiniowaną przez rozszerzenie jego podmiotu i intencję jego predykatu.
- Ponieważ elementy wiedzy w danych wejściowych mogą pochodzić z różnych źródeł, mogą zawierać niespójności.
- Gdy pojawia się nowa wiedza, zazwyczaj system nie może sobie pozwolić na czas, aby odpowiednio zaktualizować wszystkie poprzednie przekonania.

Dlatego, chociaż każde stwierdzenie można traktować jako osąd probabilistyczny, różne stwierdzenia odpowiadają różnym przestrzeniom dowodowym, a ich wartości prawdy są oceniane w odniesieniu do różnych zbiorów dowodów. W rezultacie odpowiadają one różnym rozkładom prawdopodobieństwa. Na przykład, jeśli traktujemy częstotliwość jako prawdopodobieństwo, reguła dedukcji powinna obliczyć  $Pr_3(S \rightarrow P)$  z  $Pr_1(M \rightarrow P)$  i  $Pr_2(S \rightarrow M)$ . W standardowej teorii prawdopodobieństwa istnieje niewiele wyników, które można zastosować do tego rodzaju obliczeń rozkładu krzyżowego. NARS rozwiązuje ten problem, wykraczając poza teorię prawdopodobieństwa, choć nadal dzieląc z nią pewną intuicję i wynik [43]. W NARS ilość dowodów jest definiowana w taki sposób, że można jej użyć do wskazania losowości, niejasności i niewiedzy w celu porównania z teorią Dempstera-Shafera. Chociaż różne rodzaje niepewności mają różne pochodzenie, zwykle współistnieją i są ze sobą splątane w sytuacjach praktycznych. Ponieważ NARS nie nakłada żadnych ograniczeń na to, co może się wydarzyć w jego doświadczeniu, i musi podejmować uzasadnione decyzje, gdy dostępna wiedza jest niewystarczająca, taki ujednoczony pomiar niepewności jest konieczny. W NARS mogą występować konflikty przekonań, w tym sensie, że temu samemu stwierdzeniu przypisuje się różne wartości prawdy, gdy pochodzi z różnych części doświadczenia. Przy niewystarczających zasobach NARS nie może znaleźć i wyeliminować wszystkich ukrytych konfliktów w swojej bazie wiedzy. To, co może zrobić, to wygenerować, gdy zostanie znaleziony konflikt, podsumowany wniosek, którego wartość prawdy odzwierciedla połączone dowody. Te konflikty są normalne, a nie wyjątkowe. W rzeczywistości ich istnienie jest główną siłą napędową uczenia się i tylko dzięki ich rozwiązaniom niektóre typy wnioskowania, takie jak indukcja i abdukcja, mogą mieć swoje wyniki akumulowane [39]. W logice predykatów pierwszego rzędu para sprzecznych propozycji implikuje wszystkie propozycje. Nie dzieje się tak w logice terminów takiej jak NARS, ponieważ w logice terminów wnioski i przesłanki muszą mieć wspólne terminy, a stwierdzenia o tej samej wartości logicznej nie mogą się nawzajem zastępować w wyprowadzeniu (jak ma to miejsce w logice predykatów). W rezultacie NARS toleruje implicate

sprzeczne przekonania i rozwiązuje jawne konflikty poprzez łączenie dowodów. Koncepcje w NARS są niepewne, ponieważ znaczenie koncepcji nie jest określone przez interpretację, która łączy ją z obiektem zewnętrznym, ale przez jej relacje z innymi koncepcjami. Relacje są z kolei określone przez doświadczenie systemu i jego przetwarzanie doświadczenia. Kiedy koncepcja jest zaangażowana w przetwarzanie zadania, zwykle wykorzystywana jest tylko część wiedzy związanej z koncepcją. W konsekwencji koncepcje stają się „płynne”:

1. Żadne pojęcie nie ma jasno określonej granicy. To, czy pojęcie jest instancją innego pojęcia, jest kwestią stopnia. Dlatego wszystkie pojęcia w NARS są „niejasne”.
2. Oceny członkostwa są rewidowalne. Podział priorytetów między relacjami od pojęcia do innych pojęć również zmienia się od czasu do czasu. Dlatego to, co pojęcie faktycznie oznacza dla systemu, jest zmienne.
3. Jednak znaczenie pojęcia nie jest arbitralne ani losowe, ale stosunkowo stabilne, ograniczone doświadczeniem systemu.

### **NARS jako sieć równoległa i rozproszona**

Chociaż wszystkie poprzednie opisy przedstawiają NARS jako system rozumowania z formalnym językiem i regułami, w rzeczywistości system ten można również opisać jako sieć. Możemy postrzegać każdy termin jako węzeł, a każde stwierdzenie jako łącze między dwoma węzłami, a odpowiadającą mu wartość logiczną jako siłę łącza. Priorytety są definiowane między węzłami i łączami. Na każdym etapie wnioskowania dwa sąsiednie łącza generują nowe łącza, a różne typy wnioskowania odpowiadają różnym kombinacjom łączy. Odpowiedź na pytanie oznacza określenie siły łącza, biorąc pod uwagę jego węzeł początkowy i końcowy, lub zlokalizowanie węzła z najsilniejszym łączem od lub do danego węzła. Ponieważ poprzez stosowanie reguł zmienia się struktura topologiczna sieci, siła łączy i rozkład priorytetów, to, co robi system, to znacznie więcej niż przeszukiwanie statycznej sieci w celu znalezienia pożądanego łącza lub węzła. Zgodnie z taką interpretacją NARS wykazuje pewne podobieństwo do innych podejść AI opartych na sieci, takich jak modele koneksjonistyczne. W NARS wiele procesów współistnieje jednocześnie. System nie tylko przetwarza zadania wejściowe równoległe, ale także wykonuje je w przypadku podzadań pochodnych. Fakt, że system można zaimplementować w maszynie jednoprocessorowej, nie zmienia sytuacji, ponieważ nie chodzi tu o to, aby procesy działały dokładnie w tym samym czasie na kilku elementach sprzętowych (choć możliwe jest zaimplementowanie NARS w systemie wieloprocessorowym), ale o to, aby nie były uruchamiane jeden po drugim, tzn. jeden proces zaczyna się po zakończeniu drugiego. Taki model przetwarzania równoległego jest przyjmowany przez NARS, ponieważ biorąc pod uwagę niewystarczalność wiedzy i zasobów, a także dynamiczną naturę struktury pamięci i konkurencję o zasoby, niemożliwe jest, aby system przetwarzał zadania jedno po drugim. Wiedza w NARS jest reprezentowana rozproszona w tym sensie, że nie ma bezpośredniej korespondencji między wejściem/wyjściem w doświadczeniu/odpowiedzi a wiedzą w pamięci [12]. Kiedy systemowi dostarczany jest nowy element wiedzy, nie jest on po prostu wstawiany do pamięci. Będą się zdarzać spontaniczne wnioski, które generują wnioski pochodne. Ponadto, nowa wiedza może zostać zrewidowana, gdy jest w konflikcie z poprzednią wiedzą. W rezultacie, pojawienie się nowej wiedzy może powodować nielocalne efekty w pamięci. Z drugiej strony, odpowiedź na pytanie może być generowana przez nielocalną wiedzę. Na przykład, odpowiadając na pytanie „Czy gołąb jest rodzajem ptaka?”, część wiedzy „gołąb → ptak” (z jej wartością logiczną) przechowywana w koncepcjach gołąb i ptak dostarcza gotową odpowiedź, ale praca się nie zatrzymuje. Podzadania są generowane (z niższym priorytetem) i wysyłane do powiązanych koncepcji. Ponieważ mogą występować ukryte konflikty w bazie wiedzy, poprzednia „lokalna” odpowiedź może zostać zrewidowana przez wiedzę przechowywaną gdzie indziej. Dlatego

też przyswajanie nowej wiedzy i generowanie odpowiedzi są zarówno nielokalnymi zdarzeniami w pamięci, chociaż koncepcje odpowiadające terminom, które pojawiają się bezpośrednio w wiedzy wejściowej/pytaniu, zwykle mają większy wkład. Jak „globalne” może być takie zdarzenie, jest określane zarówno przez dostępną wiedzę, jak i zasoby przydzielone do zadania. W NARS informacje nie są tylko przechowywane rozproszone i z duplikacjami, ale także przetwarzane przez wiele ścieżek. Przy niewystarczającej wiedzy i zasobach, gdy zadawane jest pytanie lub przekazywana jest część wiedzy, zwykle nie można zdecydować, czy spowoduje to redundancję lub jaka jest najlepsza metoda jej przetworzenia, więc wiele kopii i ścieżek staje się nieuniknionych. Redundancja może pomóc systemowi odzyskać się po częściowym uszkodzeniu, a także sprawić, że zachowania systemu będą zależne od zdarzeń statystycznych. Na przykład, jeśli to samo pytanie jest zadawane wielokrotnie, otrzyma więcej czasu procesora. W przeciwieństwie do wielu symbolicznych systemów AI, NARS nie jest „kruchy” - to znaczy, że nie można go łatwo „zabić” przez niewłaściwe dane wejściowe. NARS jest otwarty i niezależny od domeny, więc każda wiedza i pytanie, o ile można je wyrazić w języku interfejsu systemu, mogą zostać zaakceptowane przez system. Konflikt między nową wiedzą a poprzednią wiedzą nie spowoduje „paradoksu implikacji” (tj. z niespójności można wyprowadzić dowolne propozycje). Wszystkie błędy w wiedzy wejściowej mogą zostać skorygowane przez przyszłe doświadczenia w różnym stopniu. Pytania wykraczające poza obecne możliwości systemu nie będą już powodować „eksplozji kombinatorycznej”, ale zostaną stopniowo porzucone przez system po pewnych daremnych wysiłkach. W ten sposób system może nie odpowiedzieć na pewne pytanie, ale taka porażka nie spowoduje paraliżu. Zgodnie ze sposobem działania NARS, każde pojęcie jako jednostka przetwarzająca zajmuje się tylko swoimi własnymi sprawami, to znaczy, że wyciąga wnioski tylko tam, gdzie pojęcie jest bezpośrednio zaangażowane. W rezultacie odpowiedź na pytanie jest zwykle współpracą kilku koncepcji. Podobnie jak w modelach koneksjonistycznych, nie ma „globalnego planu” ani „centralnego procesu”, który byłby odpowiedzialny za każde pytanie. Współpraca odbywa się poprzez przekazywanie wiadomości między koncepcjami. Generowanie określonego rozwiązania jest wyłaniającym się rezultatem wielu lokalnych zdarzeń, nie tylko spowodowanych przez zdarzenia na ścieżce jego pochodzenia, ale także przez aktywność innych zadań, które dostosowują strukturę pamięci i konkurują o zasoby. Z tego powodu każde zdarzenie w NARS jest pod wpływem wszystkich zdarzeń, które mają miejsce przed nim. Z powyższych właściwości wynika bezpośrednio, że rozwiązanie określonego zadania jest zależne od kontekstu. Zależy ono nie tylko od samego zadania i wiedzy, jaką posiada system, ale również od tego, jak wiedza jest zorganizowana i jak zasoby są przydzielane w danym momencie. Kontekst, w którym system otrzymuje zadanie, czyli to, co dzieje się przed i po zadaniu w doświadczeniu systemu, silnie wpływa na to, jakie rozwiązanie otrzymuje zadanie. Dlatego jeśli system otrzyma to samo zadanie dwa razy, rozwiązania mogą być (choć niekoniecznie) różne, nawet jeśli w tym czasie systemowi nie zostanie przekazana żadna nowa wiedza. Tutaj „kontekst” oznacza bieżące środowisko pracy, w którym zadanie jest przetwarzane. Takie konteksty są dynamiczne i ciągłe, a nie są to z góry określone sytuacje indeksowane etykietami takimi jak „bank” i „hotel”.

### **Konkurencja zasobów**

System nie traktuje wszystkich procesów równo. Rozdziela zasoby między procesy i pozwala każdemu z nich postępować tylko z określoną prędkością i do określonej „głębokości” w bazie wiedzy, w zależności od tego, ile zasobów jest dostępnych dla systemu. Również z powodu niewystarczającej wiedzy dystrybucja zasobów jest utrzymywana dynamicznie (dostosowywana w trakcie działania procesów), a nie statycznie (planowana przed rozpoczęciem działania procesów), ponieważ dystrybucja zależy od sposobu ich działania. W rezultacie procesy konkurują ze sobą o zasoby. Przyspieszenie jednego procesu oznacza spowolnienie innych. Wartość priorytetu zadania odzwierciedla jego (względny) priorytet w konkurencji, ale nie określa jego (bezwzględnego)

rzeczywistego zużycia zasobów, które zależy również od wartości priorytetów innych współistniejących zadań. Przy niewystarczającym czasie przetwarzania nieefektywne jest równe traktowanie całej wiedzy i pytań. W NARS niektóre z nich (o wyższych wartościach priorytetu) otrzymują więcej uwagi, to znaczy są bardziej aktywne lub dostępne, podczas gdy inne są tymczasowo zapomniane. Przy niewystarczającej ilości miejsca w pamięci, część wiedzy i pytań zostanie trwale zapomniana — usunięta z pamięci. Podobnie jak w ludzkiej pamięci, w NARS zapominanie nie jest działaniem celowym, ale efektem ubocznym spowodowanym przez konkurencję o zasoby. W tradycyjnych systemach komputerowych ilość czasu spędzonego na zadaniu jest określana przez projektanta systemu, a użytkownik dostarcza zadania w czasie wykonywania bez wymagań czasowych. Z drugiej strony, wiele systemów czasu rzeczywistego pozwala użytkownikom na dołączenie terminu do zadania, a czas spędzony na zadaniu jest określany przez termin. Odmianą tego podejścia jest to, że zadanie jest dostarczane bez terminu, ale użytkownik może przerwać proces w dowolnym momencie, aby uzyskać najlepszą jak dotąd odpowiedź. NARS wykorzystuje bardziej elastyczną metodę decydowania, ile czasu poświęcić na zadanie, a zarówno system, jak i użytkownik wpływają na decyzję. Użytkownik może dołączyć początkową wartość priorytetu do zadania, ale rzeczywisty przydział zależy również od bieżącej sytuacji systemu, a także od tego, jak dobrze przebiega przetwarzanie zadania. W rezultacie to samo zadanie, o tym samym priorytecie początkowym, będzie przetwarzane częściej, gdy system jest „bezczynny”, niż gdy jest „zajęty”.

### **Elastyczne zachowania**

W NARS sposób generowania odpowiedzi w dużej mierze zależy od tego, jaka wiedza jest dostępna i jak jest zorganizowana. Stając przed zadaniem, system nie wybiera najpierw metody, a następnie odpowiednio zbiera wiedzę, ale pozwala jej na interakcję z dostępną wiedzą. Na każdym etapie wnioskowania metoda używana do przetwarzania zadania jest określana przez rodzaj wiedzy, która jest w danym momencie pobierana. W rezultacie ścieżka przetwarzania dla zadania jest określana dynamicznie w czasie wykonywania, przez bieżącą strukturę pamięci i dystrybucję zasobów systemu, a nie przez z góry ustalony algorytm zorientowany na problem. Zasadniczo zachowanie NARS jest nieprzewidywalne na podstawie samego zadania wejściowego, chociaż nadal przewidywalne na podstawie początkowego stanu systemu i pełnego doświadczenia. W celach praktycznych zachowanie NARS nie jest dokładnie przewidywalne dla ludzkiego obserwatora. Aby dokładnie przewidzieć rozwiązanie systemu dla określonego zadania, obserwator musi znać wszystkie szczegóły początkowego stanu systemu i uważnie śledzić doświadczenie systemu, aż do momentu faktycznego wygenerowania rozwiązania. Gdy system jest wystarczająco złożony (w porównaniu ze zdolnością przetwarzania informacji przez predyktora), nikt nie może tego zrobić. Nie oznacza to jednak, że system działa w sposób losowy. Jego zachowania są nadal determinowane przez stan początkowy i doświadczenie, więc możliwe są przybliżone przewidywania. Jeśli NARS zostanie zaimplementowany w komputerze von Neumanna, czy może wyjść poza zakres informatyki? Tak, jest to możliwe, ponieważ system komputerowy jest hierarchią o wielu poziomach. Niektórzy krytycy zakładają implícite, że ponieważ pewien poziom systemu komputerowego może zostać uchwycony przez logikę predykatów pierwszego rzędu i maszynę Turinga, teorie te wiążą również wszystkie wydajności, jakie system może mieć. Tak nie jest. Gdy system A jest implementowany przez system B, ten pierwszy niekoniecznie dziedziczy wszystkie właściwości tego drugiego. Na przykład nie możemy powiedzieć, że komputer nie może przetwarzać liczb dziesiętnych (ponieważ są implementowane przez liczby binarne), nie może przetwarzać symboli (ponieważ są kodowane przez cyfry) lub nie może używać języka programowania funkcyjnego lub logicznego (ponieważ są one ostatecznie tłumaczone na proceduralny język maszynowy). Oczywiście jest, że NARS, ze swoimi płynnymi koncepcjami, rewidowalną wiedzą i omylnymi regułami wnioskowania, łamie zasady klasycznej logiki. Jednak jako maszyna wirtualna, NARS może opierać się na innej maszynie wirtualnej, która jest systemem czysto aksjomatycznym, jak

pokazuje praktyka jej implementacji, a ten fakt nie czyni systemu „aksjomatycznym”. Jeśli przyjmiemy kompletne doświadczenie i odpowiedź systemu jako dane wejściowe i wyjściowe, to NARS jest nadal maszyną Turinga, która zdecydowanie mapuje dane wejściowe na dane wyjściowe w skończonych krokach. To, co się tu dzieje, zostało wskazane przez Hofstadtera jako „coś może być obliczeniowe na jednym poziomie, ale nie na innym”, a przez Kugła jako „procesy poznawcze, które, chociaż obejmują coś więcej niż obliczenia, nadal mogą być modelowane na maszynach, które nazywamy „komputerami”” [18]. Wręcz przeciwnie, tradycyjne systemy komputerowe są maszynami Turinga albo globalnie (od doświadczenia do odpowiedzi), albo lokalnie (od pytania do odpowiedzi).

### **Autonomia i kreatywność**

Globalne zachowanie NARS jest determinowane przez „wypadkową siłę” jego wewnętrznych zadań. Początkowo system jest napędzany tylko przez zadania wejściowe. Następnie system rekurencyjnie wywodzi podzadania, stosując reguły wnioskowania do zadań i dostępnej wiedzy. Nie ma jednak gwarancji, że osiągnięcie zadań pochodnych okaże się naprawdę pomocne lub nawet powiązane z zadaniami oryginalnymi, ponieważ wiedza, na której opiera się wyprowadzenie, jest rewidowalna. Z drugiej strony, niemożliwe jest, aby system zawsze prawidłowo określił, które zadania są bliżej powiązane z zadaniami oryginalnymi. W rezultacie zachowanie systemu będzie w pewnym stopniu zależało od „jego własnych zadań”, które są w rzeczywistości mniej lub bardziej niezależne od procesów oryginalnych, nawet jeśli historycznie z nich wywodzą się. Jest to zjawisko autonomii funkcjonalnej. W skrajnej formie zadania pochodne mogą stać się tak silne, że uniemożliwiają nawet realizację zadań wejściowych. W ten sposób zadania pochodne są wyobcowane. Alienacja i nieprzewidywalność czasami powodują, że system jest „poza kontrolą”, ale jednocześnie prowadzą do kreatywnych i oryginalnych zachowań, ponieważ system realizuje cele, które nie są bezpośrednio przypisane przez jego otoczenie lub jego wrodzoną naturę, za pomocą metod, które nie są bezpośrednio wywnioskowane z danej wiedzy. Kreatywność nie oznacza, że wszystkie wyniki takich zachowań są korzystne dla systemu lub doskonałe według zewnętrznych standardów. Nie oznacza to również, że te zachowania pochodzą znikąd lub z jakiejś „wolnej woli”. Wręcz przeciwnie, oznacza to, że zachowania są nowe dla systemu i nie można ich przypisać ani projektantowi (który określa początkowy stan i umiejętności systemu), ani nauczycielowi (który określa część doświadczenia systemu). Projektanci i nauczyciele jedynie umożliwiają kreatywne zachowania. To, co zamienia możliwość w rzeczywistość, to doświadczenie systemu, a w przypadku systemu, który żyje w złożonym środowisku, jego doświadczenie nie jest całkowicie zdeterminowane przez żadne inne systemy (ludzkie lub komputerowe). Z tego powodu te zachowania, wraz z ich wynikami, lepiej przypisać samemu systemowi, niż komukolwiek innemu.

Tradycyjne systemy komputerowe zawsze powtarzają następujący „cykl życia”:

- oczekiwanie na zadania
- akceptowanie zadania
- praca nad zadaniem zgodnie z algorytmem
- zgłaszanie rozwiązania dla zadania
- oczekiwanie na zadania
- . . .

Wręcz przeciwnie, NARS ma „własny czas życia”. Gdy system jest wystarczająco doświadczony, będzie miał wiele zadań do przetworzenia. Z drugiej strony, nowe dane wejściowe mogą pojawić się w

dowolnym momencie. W rezultacie historia systemu nie jest już taka jak poprzednia pętla. System zazwyczaj pracuje nad swoimi „własnymi” zadaniami, ale jednocześnie jest zawsze gotowy do reagowania na nowe zadania dostarczane przez środowisko. Każdy element danych wejściowych zazwyczaj przyciąga uwagę systemu na jakiś czas i powoduje również pewne długoterminowe skutki. System nigdy nie osiąga „stanu końcowego” i tam się zatrzymuje, chociaż może zostać zresetowany przez użytkownika do stanu początkowego. W ten sposób każda czynność przetwarzania zadań jest częścią doświadczenia życiowego systemu i jest pod wpływem innych czynności. W porównaniu z NARS, tradycyjne systemy komputerowe traktują każdą czynność rozwiązywania problemów jako oddzielny cykl życia z ustalonym z góry końcem.

## **Wnioski**

Kluczowa różnica między NARS a głównymi projektami AI nie leży w szczegółach technicznych, ale w stanowisku filozoficznym i metodologicznym. Projekt NARS nie ma na celu rozwiązania konkretnego problemu praktycznego lub funkcji poznawczej, ale próbuje zbudować inteligentny system ogólnego przeznaczenia poprzez zidentyfikowanie „istoty inteligencji”, tj. podstawowej zasady przetwarzania informacji, a następnie zaprojektowanie komponentów systemu odpowiednio. Jak opisano powyżej, w projekcie NARS zakłada się, że „inteligencja” oznacza „adaptację przy niewystarczającej wiedzy i zasobach”, a następnie wybiera się system rozumowania jako ramy do zastosowania tego założenia. Podczas projektowania systemu odkryliśmy, że wszystkie istotne tradycyjne teorie (w tym logika predykatów pierwszego rzędu, teoria modeli, teoria prawdopodobieństwa, teoria obliczalności, teoria złożoności obliczeniowej, ...) są sprzeczne z powyższym założeniem, więc wszystkie główne komponenty muszą zostać przeprojektowane. Komponenty te, choć technicznie proste, zasadniczo różnią się od tradycyjnych komponentów w naturze. Zbudowany w ten sposób NARS zapewnia ujednoczony model wielu zjawisk obserwowanych w ludzkim poznaniu. Osiąga się to nie poprzez wyraźne dopasowanie danych psychologicznych, ale poprzez ich odtworzenie z prostego i ujednoczonego fundamentu. W ten sposób widzimy, że zjawiska te mają wspólne wyjaśnienie funkcjonalne, a wszystkie inteligentne systemy, zarówno naturalne, jak i sztuczne, będą wykazywać te zjawiska, o ile będą systemami adaptacyjnymi pracującymi z niewystarczającą wiedzą i zasobami. Projekt NARS rozpoczął się w 1983 roku na Uniwersytecie Pekinśkim. Zbudowano kilka działających prototypów w sposób przyrostowy (tj. każdy z większą liczbą reguł wnioskowania i bardziej skomplikowanym mechanizmem sterowania). Obecnie ukończono wnioskowanie pierwszego rzędu, a wnioskowanie wyższego rzędu jest w trakcie opracowywania. Chociaż cały projekt jest jeszcze daleki od ukończenia, wcześniejsze doświadczenia wykazały wykonalność tego podejścia. Aby uzyskać aktualne informacje o projekcie oraz najnowsze publikacje i demonstracje, odwiedź stronę <http://www.cogsci.indiana.edu/farg/peiwang/papers.html>.