

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



هوش مصنوعی

فصل ۱۰

بازنمایی دانایی

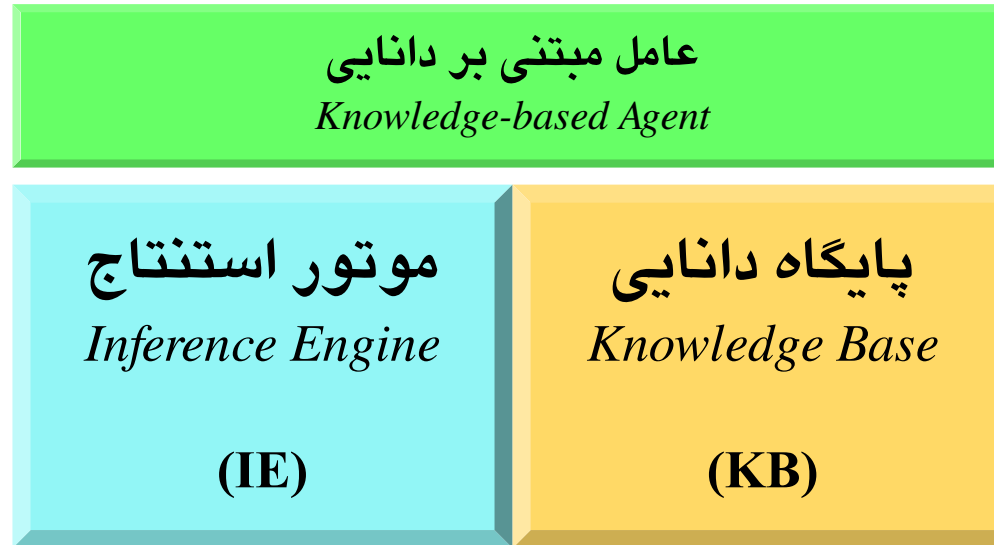
Knowledge Representation

کاظم فولادی قلعه
دانشکده مهندسی، دانشکدگان فارابی
دانشگاه تهران

<http://courses.fouladi.ir/ai>

عامل مبتنی بر دانایی

نقش بازنمایی دانایی



بازنمایی دانایی:

چگونگی بازنمایی واقعیت‌های دنیا در

پایگاه دانایی

(محتوای درون پایگاه دانایی)

هوش مصنوعی

بازنمایی دانایی

۱

مهندسی
هست شناختی

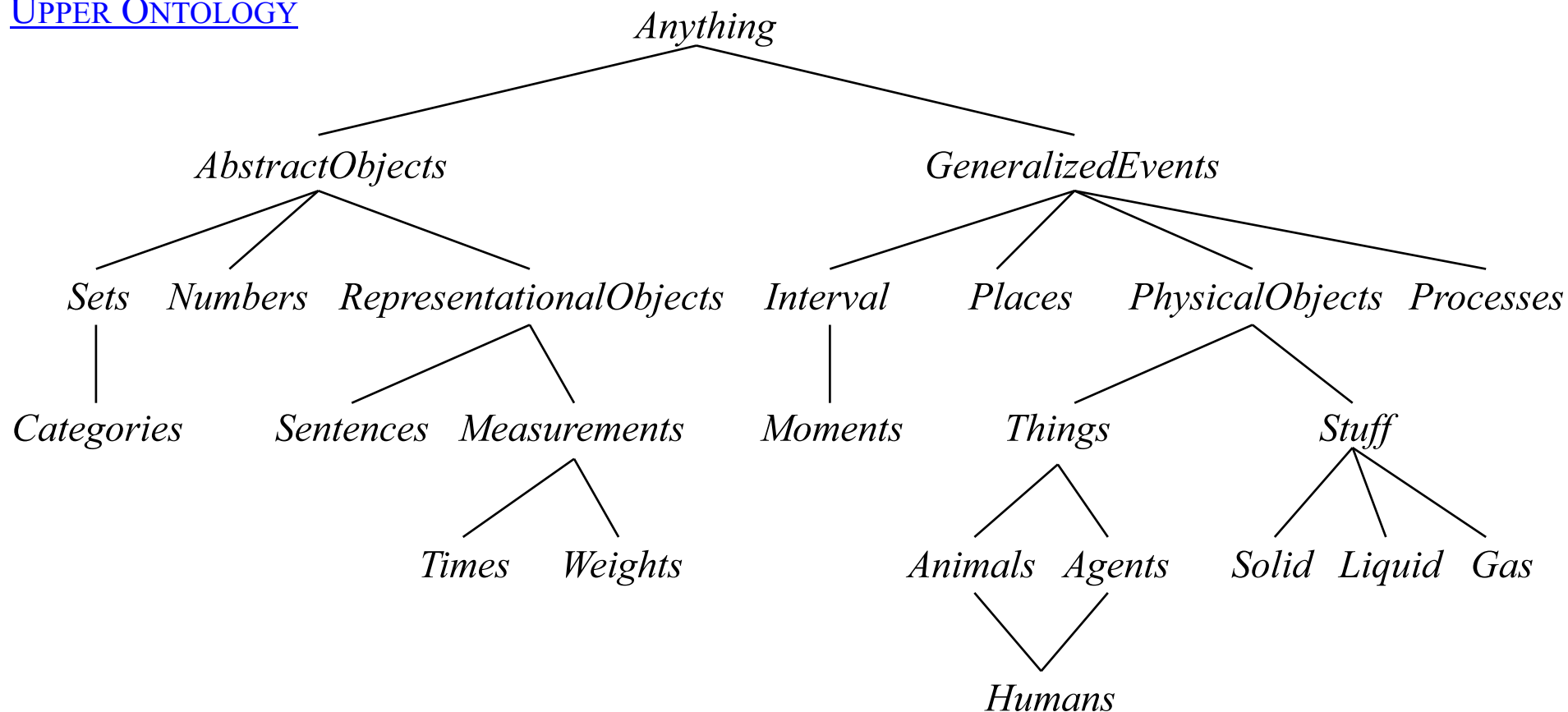
مهندسی هست شناختی

ONTOLOGICAL ENGINEERING

بازنمایی مفاهیم انتزاعی (نظیر: کنش‌ها، زمان، اشیای فیزیکی و باورها)

مهندسی هست شناختی
Ontological Engineering

هست‌شناسی بالایی

UPPER ONTOLOGY

آنچه در دنیا وجود دارد!
مفهوم پایین‌تر، حالت خاصی از مفهوم بالاتر است.

هست‌شناسی بالایی
Upper Ontology

بازنمایی دانایی

نقش منطق مرتبه اول

استفاده از منطق مرتبه اول برای سازمان‌دهی پایگاه دانایی

جنبه‌هایی از دنیای واقعی وجود دارد که توصیف آنها با FOL مشکل است:

عدم اطمینان

Uncertainty

استثناها

Exceptions

مثال:

قاعده‌ی «گوجه‌فرنگی قرمز است» استثنا دارد.

هست‌شناسی

همه‌منظوره / خاص‌منظوره

هست‌شناسی	
هست‌شناسی خاص‌منظوره <i>Special-Purpose Ontology</i>	هست‌شناسی همه‌منظوره <i>General-Purpose Ontology</i>
	دو مشخصه‌ی اصلی: - در همه‌ی دامنه‌های خاص‌منظوره، باید کم و بیش قابل به‌کارگیری باشد. (با اضافه کردن اصول موضوع خاص آن دامنه) - برای دامنه‌های به‌اندازه‌ی کافی دشوار، باید حوزه‌های مختلف دانایی را متحد کند. (زیرا استدلال و حل مسئله زمینه‌های مختلف را همزمان درگیر می‌کند)

۲

دسته‌ها و اشیا

دسته‌ها و اشیا

CATEGORIES AND OBJECTS

دسته = مجموعه‌ای از اعضای آن

دسته
Category

ضرورت سازمان‌دهی اشیا در دسته‌ها در بازنمایی دانایی

استدلال در سطح دسته‌ها
Reasoning at the level of categories

تعامل در سطح اشیا
Interaction at the level of the objects

نقش دسته‌ها در پیش‌بینی در مورد اشیا: بر اساس خواص درک شده از شیء

دسته‌ها و اشیا

بازنمایی دسته‌ها با منطق مرتبه اول

روش‌های بازنمایی دسته‌ها با منطق مرتبه اول

جسمیت‌بخشی دسته‌ها در اشیا

Reification of categories into objects

در قالب محمول‌ها

Predicates

Apples

Apple(x)

دسته‌ها و اشیا

سازمان‌دهی دسته‌ها با رابطه‌ی وراثت

وراثت
Inheritance

رابطه‌ی بین دسته‌ها

مثال:

همه‌ی نمونه‌های خوراکی‌ها خوردنی هستند،
میوه زیرطبقه‌ای از خوراکی، و
سیب زیرطبقه‌ای از میوه است؛
پس: سیب خوردنی است

رابطه‌ی وراثت بین دسته‌ها، یک رده‌بندی (*taxonomy*) تعریف می‌کند.

دسته‌ها و منطق مرتبه اول

مثال

یک شیء عضوی از یک دسته است.

An object is a member of a category

$$MemberOf(BB_{12}, Basketballs)$$

یک دسته یک زیرطبقه از یک دسته‌ی دیگر است.

A category is a subclass of another category

$$SubsetOf(Basketballs, Balls)$$

همه‌ی عضوهای یک دسته، تعدادی خصوصیت دارند.

All members of a category have some properties

$$\forall x (MemberOf(x, Basketballs) \Rightarrow Round(x))$$

همه‌ی عضوهای یک دسته می‌توانند توسط برخی خصوصیات بازشناسی شوند.

All members of a category can be recognized by some properties

$$\forall x (Orange(x) \wedge Round(x) \wedge Diameter(x) = 9.5in \wedge MemberOf(x, Balls) \Rightarrow MemberOf(x, Basketballs))$$

یک دسته به عنوان یک کل تعدادی خصوصیت دارد.

A category as a whole has some properties

$$MemberOf(Dogs, DomesticatedSpecies)$$

روابط بین دسته‌ها

رابطه‌ی مجزا بودن

مجزا بودن
Disjoint

دو یا چند دسته مجزا هستند، اگر هیچ عضو مشترکی نداشته باشند.

Two or more categories are **disjoint** if they have no members in common:

$$Disjoint(s) \Leftrightarrow$$

$$(\forall c_1, c_2 \quad c_1 \in s \wedge c_2 \in s \wedge c_1 \neq c_2 \Rightarrow Intersection(c_1, c_2) = \{\})$$

Example: $Disjoint(\{Animals, Vegetables\})$

روابط بین دسته‌ها

رابطه‌ی تجزیه‌ی جامع

مجموعه‌ای از دسته‌ها s تشکیل یک تجزیه‌ی جامع از یک دسته‌ی c می‌دهند اگر همه‌ی عضوهای مجموعه c توسط دسته‌های s پوشش داده شوند.

تجزیه‌ی جامع
Exhaustive Decomposition

A set of categories s constitutes an **exhaustive decomposition** of a category c if all members of the set c are covered by categories in s :

$$ExhaustiveDecomposition(s, c) \Leftrightarrow (\forall i \quad i \in c \Leftrightarrow \exists c_2 \quad c_2 \in s \wedge i \in c_2)$$

Example:

ExhaustiveDecomposition(\{Americans, Canadian, Mexicans\}, NorthAmericans).

روابط بین دسته‌ها

افراز

یک افراز، یک تجزیه‌ی جامع مجزا است.

افراز

Partition

A **partition** is a disjoint exhaustive decomposition:

$$Partition(s, c) \Leftrightarrow Disjoint(s) \wedge ExhaustiveDecomposition(s, c)$$

Example: $Partition(\{Males, Females\}, Persons)$.

Example: Is $(\{Americans, Canadian, Mexicans\}, NorthAmericans)$ a partition? – No! There might be dual citizenships.

تعریف کردن یک دسته

افراز

دسته‌ها می‌توانند با تدارک شرایط لازم و کافی برای عضویت **تعریف** شوند.

Categories can be defined by providing necessary and sufficient conditions for membership

$$\forall x \quad Bachelor(x) \Leftrightarrow Male(x) \wedge Adult(x) \wedge Unmarried(x)$$

گونه‌های طبیعی

تعریف‌های نوعی

دسته‌ای که تعریف صریح و قاطع ندارد!

گونه‌ی طبیعی

Natural Kind

مثال: صندلی، بوته، کتاب، گوجه‌فرنگی، ...

گونه‌های طبیعی را به صورت **نوعی** تعریف می‌کنیم.**Example:** Tomatoes: sometimes green, red, yellow, black. Mostly round**One solution:** subclass using category $Typical(Tomatoes)$.

$$Typical(c) \subseteq c$$

$$\forall x \quad x \in Typical(Tomatoes) \Rightarrow Red(x) \wedge Spherical(x).$$

می‌توان واقعیت‌های مفیدی را در مورد دسته‌ها نوشت، بدون اینکه تعریف‌های دقیق داشته باشیم.

ترکیب فیزیکی

رابطه‌ی «جزئی از»

ترکیب فیزیکی یعنی: یک شیء جزئی از یک شیء دیگر است.

ترکیب فیزیکی
Physical Composition

One object may be part of another:

$PartOf(Bucharest, Romania)$

$PartOf(Romania, EasternEurope)$

$PartOf(EasternEurope, Europe)$

چون رابطه‌ی $PartOf$ تراگذری است، نتیجه می‌شود:

$PartOf(Bucharest, Europe)$

خواص بازتابی و تراگذری رابطه‌ی $PartOf$

$\forall x \quad PartOf(x, x)$

$\forall x, y, z \quad PartOf(x, y) \wedge PartOf(y, z) \Rightarrow PartOf(x, z)$

ترکیب فیزیکی

رابطه‌ی «جزئی از»: مثال

ترکیب فیزیکی، اغلب با بیان روابط ساختاری میان اجزا مشخص می‌شود.

E.g. $Biped(a) \Rightarrow$

مثال: موجود دو پا

$$\begin{aligned}
 & (\exists l_1, l_2, b) (Leg(l_1) \wedge Leg(l_2) \wedge Body(b) \wedge \\
 & PartOf(l_1, a) \wedge PartOf(l_2, a) \wedge PartOf(b, a) \wedge \\
 & Attached(l_1, b) \wedge Attached(l_2, b) \wedge \\
 & l_1 \neq l_2 \wedge (\forall l_3) (Leg(l_3) \Rightarrow (l_3 = l_1 \vee l_3 = l_2)))
 \end{aligned}$$

ترکیب فیزیکی

رابطه‌ی «افراز اجزا»

افراز یک ترکیب فیزیکی به اجزای آن
(رابطه‌ای قابل مقایسه با افراز برای دسته‌ها)

افراز اجزا
PartPartition

ترکیب فیزیکی

رابطه‌ی «بانچ»

بانچ
BunchOf

$BunchOf(X)$ یک شیء مرکب که از همه‌ی X ها تشکیل شده است.

$BunchOf(X)$: a composite object consisting of all X 's

$BunchOf(\{Apple_1, Apple_2, Apple_3\})$

$$BunchOf(\{x\}) = x$$

تعریف $BunchOf$ بر حسب رابطه‌ی $PartOf$:

$$\forall x \ x \in s \Rightarrow PartOf(x, BunchOf(s))$$

$BunchOf(s)$ کوچک‌ترین شیئی است که شرط زیر را ارضا می‌کند:

$$\forall y [\forall x \ x \in s \Rightarrow PartOf(x, y)] \Rightarrow PartOf(BunchOf(s), y)$$

می نیمم سازی منطقی

LOGICAL MINIMIZATION

تعریف یک شیء
به عنوان کوچک ترین چیزی که شرط مشخصی را ارضا می کند.

می نیمم سازی منطقی
Logical Minimization

مثال

$BunchOf(s)$ کوچک ترین شیئی است که شرط زیر را ارضا می کند:

$$\forall y [\forall x (x \in s \Rightarrow PartOf(x, y)) \Rightarrow PartOf(BunchOf(s), y)]$$

اندازه‌گیری‌ها

MEASUREMENTS

اشیا دارای خواصی چون ارتفاع، جرم، هزینه و ... هستند.

اشیای اندازه‌ای
Measure Objects

اندازه: مقادیری که به خواص شیئی نسبت داده می‌شود.

- برای بیان، تابع واحد با یک عدد ترکیب می‌شود: $Length(L_1) = Inches(1.5) = Centimeters(3.81)$.
- Conversion between units: $\forall i \quad Centimeters(2.54 \times i) = Inches(i)$.
برای تبدیل بین واحدها:

برخی اندازه‌ها، مقیاس (scale) ندارند: مثل زیبایی، دشواری، ...

مهم‌ترین جنبه‌ی اندازه‌ها، مقادیر عددی نیست؛ بلکه این است که اندازه‌ها رتبه‌پذیر (orderable) هستند.

بنابراین به اعداد دقیق واقعی خیلی توجهی نداریم!

فیزیک کیفی

QUALITATIVE PHYSICS

فیزیک کیفی

Qualitative Physics

یک زیرحوزه از هوش مصنوعی که بررسی می‌کند چگونه می‌توان در
مورد سیستم‌های فیزیکی استدلال کرد
بدون اینکه درگیر معادلات مفصل و شبیه‌سازی‌های عددی بشویم.

جوهرها و اشیا

SUBSTANCES AND OBJECTS



خواص اشیا

خواص عرضی

Extrinsic Properties

خواص متعلق به شیء به عنوان یک کل

پس از تقسیم شیء تغییر می کند.

مثل: وزن، طول، شکل، کارکرد، ...

خواص ذاتی

Intrinsic Properties

خواص متعلق به جوهر تشکیل دهنده
شیء نه خود شیء به عنوان یک کل

با تقسیم شیء حفظ می شوند.

مثل: چگالی، نقطه جوش، رنگ، ...

جنس و چیز

STUFF AND THINGچیز
Thing

عمومی ترین دسته‌ی اشیای گسسته بدون
مشخص کردن هیچ خاصیت عرضی

جنس
Stuff

عمومی ترین دسته‌ی جوهرها بدون مشخص
کردن هیچ خاصیت ذاتی

تمامی اشیای فیزیکی به هر دو دسته تعلق دارند.

نام جرمی (جوهر)
Mass Noun (Substance)

طبقه‌ای از اشیا که در تعریف آنها فقط
خواص ذاتی وجود دارد.

شمارش ناپذیر

مثل: کره، آب، انرژی، ...

نام شمارشی
Count Noun

طبقه‌ای از اشیا که در تعریف آنها فقط
خواص عرضی وجود دارد.

شمارش پذیر

مثل: گربه‌ها، حفره‌ها، قضیه‌ها، ...

$x \in Butter \wedge PartOf(y, x) \Rightarrow y \in Butter$

$x \in Butter \Rightarrow MeltingPoint(x, Centigrade(30))$

هوش مصنوعی

بازنمایی دانایی

۳

رویدادها

حساب وضعیت

SITUATION CALCULUS

حساب وضعیت

Situation Calculus

راهی برای بازنمایی و پیگیری تغییرات در FOL

یک آرگومان وضعیت به هر محمول غیرابدی اضافه می‌کند.

حالت حاصل از اجرای کنش‌ها

وضعیت

Situation

بازنمایی زمان با موقعیت‌ها

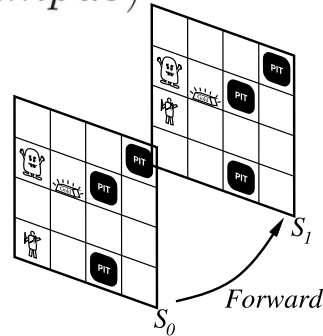
حساب وضعیت

مؤلفه‌ها

SITUATION CALCULUS

Situation calculus:

- **Actions** are logical terms
- **Situations** are logical terms consisting of
 - The initial situation $I (S_0)$
 - All situations resulting from the action on I ($= Result(a, s)$)
- **Fluents** are functions and predicates that vary from one situation to the next.
E.g. $\neg Holding(G_1, S_0)$
- **Atemporal (Eternal) predicates** are also allowed

E.g. $Gold(G_1), LeftLegOf(Wumpus)$ 

حساب وضعیت

Situation Calculus

کنش‌ها

Actions

ترم‌های منطقی

وضعیت‌ها

Situations

ترم‌های منطقی

سیال‌ها

Fluents

تابع‌ها و محمول‌های متغیر از یک وضعیت به دیگری

محمول‌ها و توابع غیرموقتی (ابدی)

Atemporal (Eternal) Predicates

محمول‌ها و توابع تغییرناپذیر

حساب وضعیت

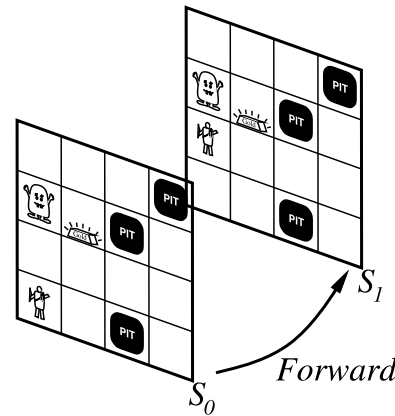
کنش‌ها و حاصل آنها

SITUATION CALCULUS

حاصل یک دنباله از کنش‌ها، توسط کنش‌های انفرادی تعیین می‌شود.

$$Result([], s) = s$$

$$Result([a \mid seq], s) = Result(seq, Result(a, s))$$



کار طرح‌ریزی

Planning Task

یافتن دنباله‌ای از کنش‌ها برای دستیابی به یک اثر مطلوب

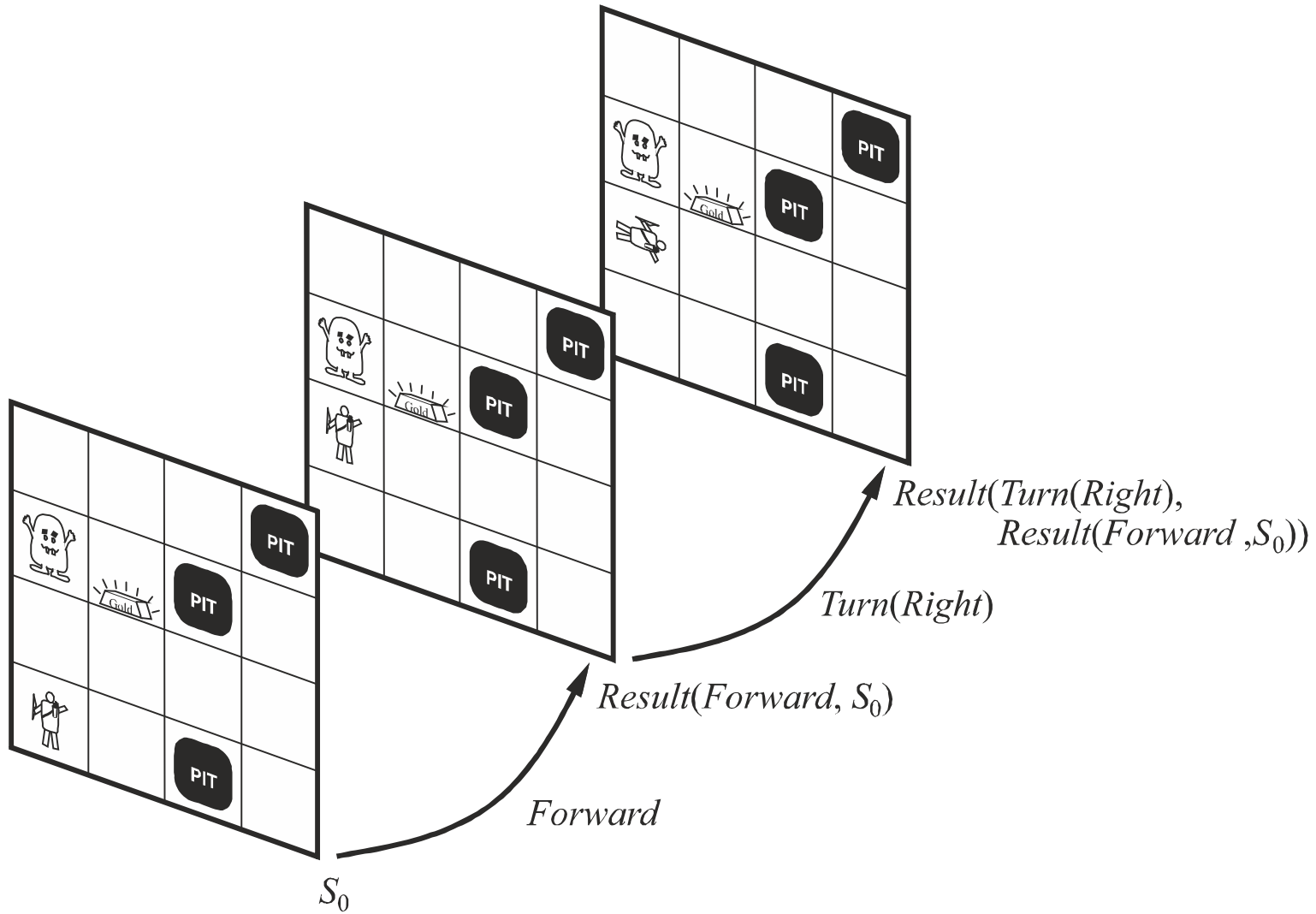
کار افکنش

Projection Task

استنباط پی‌آمد یک دنباله از کنش‌ها

حساب وضعیت

مثال: دنیای اژدها

SITUATION CALCULUS

حساب وضعیت

توصیف کنش‌ها

SITUATION CALCULUS

هر کنش در حساب وضعیت، برای توصیف تغییر دو اصل موضوع نیاز دارد:

اصل موضوع امکان

Possibility Axiom

چه زمانی انجام کنش ممکن است؟

$$At(Agent, x, s) \wedge Adjacent(x, y) \Rightarrow Poss(Go(x, y), s)$$

اصل موضوع تأثیر

Effect Axiom

توصیف تغییرات پس از انجام کنش

$$Poss(Go(x, y), s) \Rightarrow At(Agent, y, Result(Go(x, y), s))$$

مشکل اصل موضوع تأثیر: می‌گوید چه چیزهایی تغییر می‌کند، اما نمی‌گوید چه چیزهایی یکسان می‌ماند!

حساب وضعیت

مسئله‌ی قاب

FRAME PROBLEM

مسئله‌ی قاب

Frame Problem

چگونه می‌توان همه‌ی چیزهایی که بدون تغییر باقی مانده‌اند را بازنمایی کرد؟

یک راه‌حل:

اصل موضوع قاب

Frame Axiom

توصیف عدم تغییرات در نتیجه‌ی کنش‌ها

$$At(o, x, s) \wedge (o \neq Agent) \wedge \neg Holding(o, s) \Rightarrow At(o, x, Result(Go(y, z), s))$$

برای مثال: حرکتهای عامل سایر اشیای ساکن را تکان نمی‌دهد مگر آنچه عامل در دست گرفته باشد.

حساب رویداد

کار با زمان

EVENT CALCULUS

نامناسب بودن حساب وضعیت برای کنش‌هایی که دارای مدت زمانی هستند و می‌خواهیم در مورد مدت زمانی، بازه‌های زمانی و ... استدلال کنیم!

حساب رویداد

Event Calculus

سیال‌ها در نقطه‌های زمانی برقرار هستند.

$T(f, t)$	Fluent f is true at time t
$Happens(e, i)$	Event e happens over the time interval i
$Initiates(e, f, t)$	Event e causes fluent f to start to hold at time t
$Terminates(e, f, t)$	Event e causes fluent f to cease to hold at time t
$Clipped(f, i)$	Fluent f ceases to be true at some point during time interval i
$Restored(f, i)$	Fluent f becomes true sometime during time interval i

حساب رویداد

اصل موضوع

اصل موضوع حساب رویداد

Event Calculus Axiom

$$T(f, t_2) \Leftrightarrow \exists e, t \text{ Happens}(e, t) \wedge \text{Initiates}(e, f, t) \wedge (t < t_2) \\ \wedge \neg \text{Clipped}(f, t, t_2)$$

$$\text{Clipped}(f, t, t_2) \Leftrightarrow \exists e, t_1 \text{ Happens}(e, t_1) \wedge \text{Terminates}(e, f, t_1) \\ \wedge (t < t_1) \wedge (t_1 < t_2)$$

رویداد تعمیم یافته

GENERALIZED EVENT

یک پاره از جهان فضا-زمان

رویداد تعمیم یافته
Generalized Event

رویداد تعمیم یافته

محمول‌ها و تابع‌های مرتبط

GENERALIZED EVENT

SubEvent(BattleOfBritain, WorldWarII)

SubEvent(WorldWarII, TwentiethCentury)

Period(e) denotes the smallest interval enclosing an event *e*.

Intervals – chunks of space-time that include all space between two points

Duration(e) denotes the length of time of an interval, *e*.

Location(e) denotes the smallest place enclosing an event *e*.

In(e1, e2) denotes *PartOf* relationship of the spatial projection of an event.

Duration(Period(WorldWarII)) > Years(5)

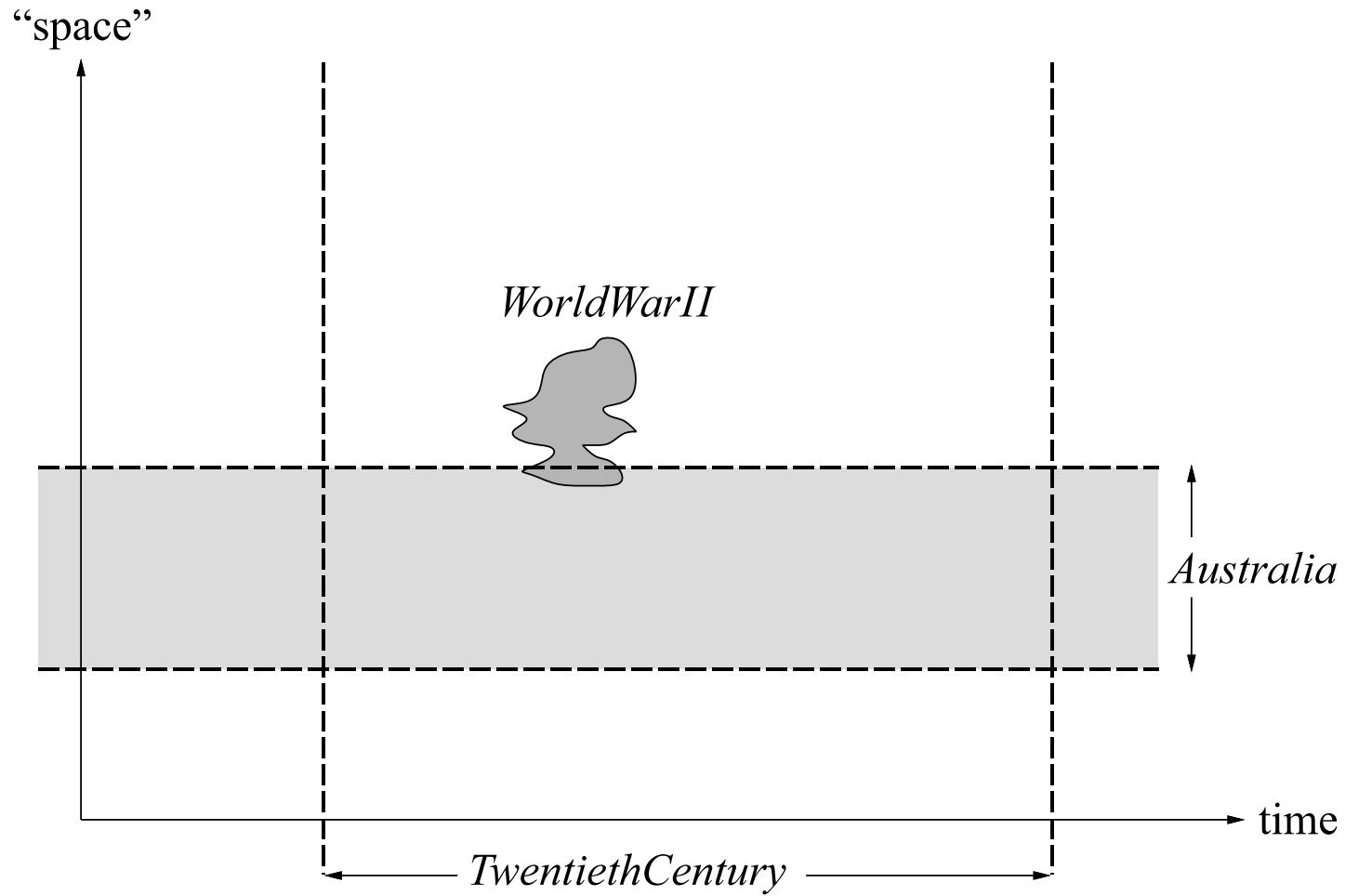
In(Sydney, Australia)

$\exists w w \in CivilWars \wedge SubEvent(w, 1640s) \wedge In(Location(w), England)$

رویداد تعمیم یافته

مثال

GENERALIZED EVENT



دسته‌ی «رویدادها»

CATEGORY OF EVENTS

کنش‌های متناظر با یک مجموعه از رویدادها (نه یک رویداد تکی)

دسته‌ی رویدادها
Category of Events

Actions like $Go([1, 1], [1, 2])$ denote a **category of events** and not single events; $Goto(y), GoFrom(x)$

کوته‌نوشت برای دسته‌های رویدادها: $E(c, i)$

$E(c, i) \Leftrightarrow \exists e \ e \in \wedge SubEvent(e, i)$

$E(Fly(Shankar, NewYork, NewDelhi), Yesterday)$

فرآیندها

PROCESSES

رویداد مایع (پیوسته)

فرآیندها
Processes

هر زیربازه از یک فرآیند، عضوی از دسته‌ی همان فرآیند است:

$E(\text{Flying}(\text{Shankar}), \text{Yesterday})$

می‌توان گفت که یک فرآیند در سراسر یک بازه جریان داشته است نه درون یک بازه:

$T(c, i) \Leftrightarrow E(c, i) \wedge \text{"the Event occurred throughout the whole interval } i\text{"}$

$T(\text{Working}(\text{Stuart}), \text{TodayLunchHour})$

فرآیندها

تمایز بین رویداد گسسته و رویداد مایع

PROCESSES

رویدادها (رویدادهای گسسته)
Events (Discrete Events)

فرآیندها (رویدادهای مایع)
Processes (Liquid Events)

جوهر فضایی
Spatial Substances

جوهر زمانی
Temporal Substances

مشابه اشیای خاص

مشابه جنس و چیز

حساب سیال

FLUENT CALCULUS

حساب سیال

Fluent Calculus

جسمیت بخشی به ترکیب سیال ها (نه سیال های منفرد)

$Both(e_1, e_2)$: the event of two things happening at once ($e_1 \circ e_2$)

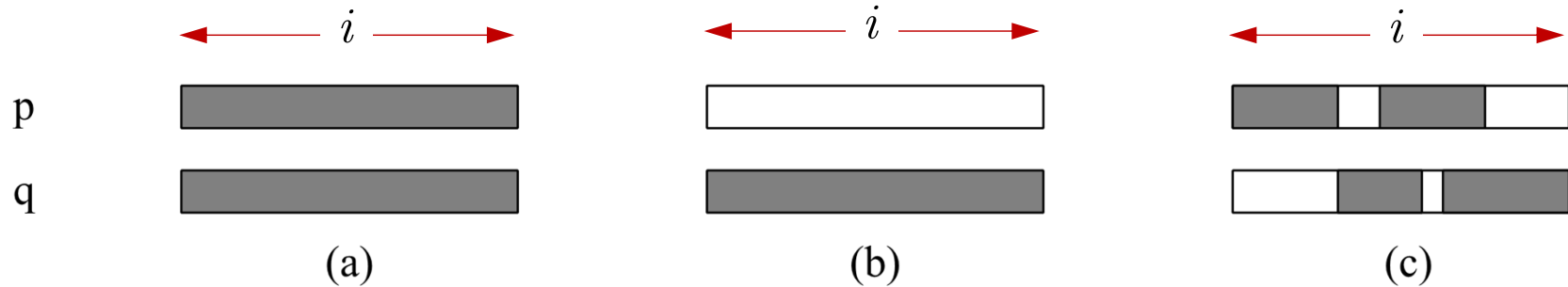
مثال: یک نفر راه می رود و همزمان آدامس می جود.

$\exists p, i (p \in People) \wedge T(Walk(p) \circ ChewGum(p), i)$

“ $\circ$ ” function is commutative and associative.

حساب سیال

رویداد ترکیبی

COMPLEEX EVENT(a) $T(Both(p, q), i)$ or $T(p \circ q, i)$ (b) $T(OneOf(p, q), i)$ (c) $T(Either(p, q), i)$

حساب سیال

بازه‌های زمانی

TIME INTERVALS

$Partition(\{Moments, ExtendedIntervals\}, Intervals)$

$i \in Moments \Leftrightarrow Duration(i) = Seconds(0)$

$Start(i)$, $End(i)$ denote the start and end moments of an interval i ;

$Interval(i) \Rightarrow Duration(i) = (Time(End(i)) - Time(Start(i)))$

Time scale: such as $Seconds(s)$

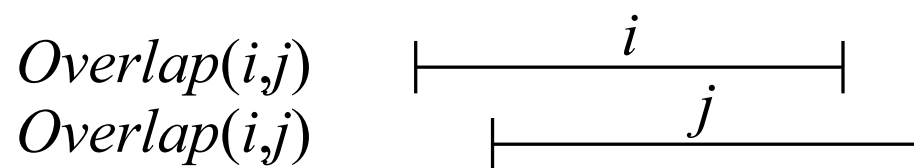
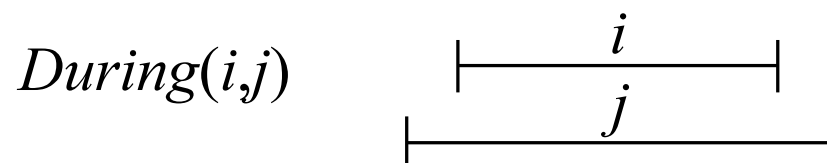
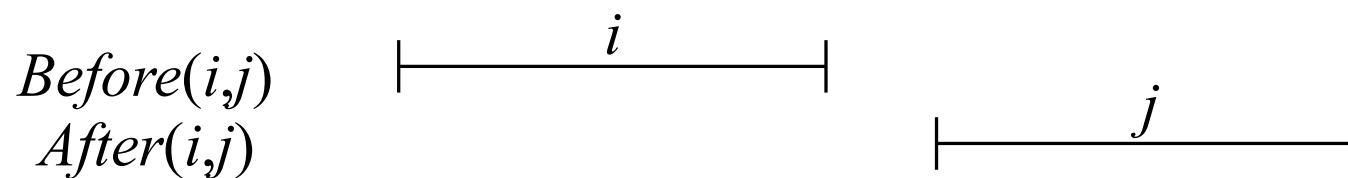
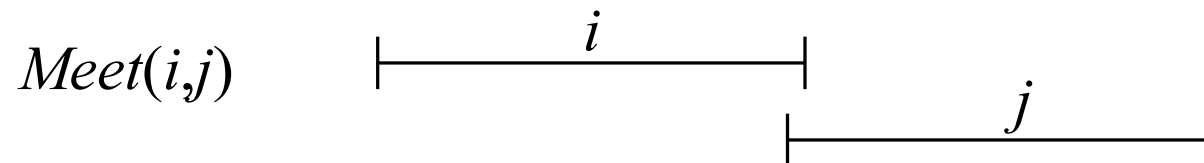
$Time(Start(AD2001)) = Date(0, 0, 0, 1, Jan, 2001)$

$Date(0, 20, 21, 24, 1, 1995) = Seconds(3000000000)$

using these constructs one can define the functions $Meet(i, j)$, $Before(i, j)$, $Overlap(i, j)$, ...

حساب سیال

محمول‌هایی روی بازه‌های زمانی

PREDICATES ON TIME INTERVALS

حساب سیال

محمول‌هایی روی بازه‌های زمانی

PREDICATES ON TIME INTERVALS

$Meet(i, j)$	$\Leftrightarrow$	$End(i) = Begin(j)$
$Before(i, j)$	$\Leftrightarrow$	$End(i) < Begin(j)$
$After(j, i)$	$\Leftrightarrow$	$Before(i, j)$
$During(i, j)$	$\Leftrightarrow$	$Begin(j) < Begin(i) < End(i) < End(j)$
$Overlap(i, j)$	$\Leftrightarrow$	$Begin(i) < Begin(j) < End(i) < End(j)$
$Begins(i, j)$	$\Leftrightarrow$	$Begin(i) = Begin(j)$
$Finishes(i, j)$	$\Leftrightarrow$	$End(i) = End(j)$
$Equals(i, j)$	$\Leftrightarrow$	$Begin(i) = Begin(j) \wedge End(i) = End(j)$

شیء فیزیکی

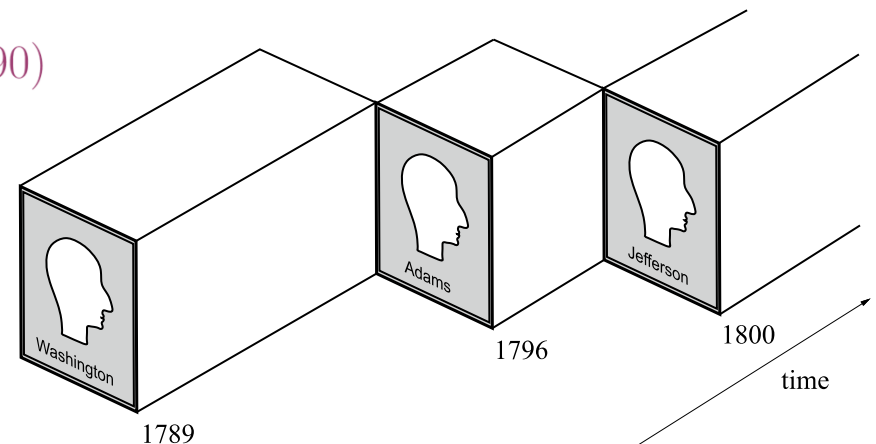
بیان خواص شیء فیزیکی با سیال‌های حالت

PHYSICAL OBJECT

تکه‌ای از فضا-زمان

(شیء فیزیکی می‌تواند به عنوان یک رویداد تعمیم‌یافته دیده شود)

شیء فیزیکی

*Physical Objects*e.g., *USA* can be thought of as an event.We can describe changing properties of *USA* using **state fluents**: $E(P_{population}(USA, 271000000), AD1999)$ *President(USA)* denotes a single object that consists of different people at different times: $T(President(USA) = GeorgeWashington, AD1790)$ = is a function symbol,
logical = is not something that can change over time.

۴

رویدادهای
ذهنی
و
اشیای
ذهنی

اشیای ذهنی - رویداد ذهنی

عامل می تواند باورهایی داشته باشد و باورهای جدیدی را استنباط کند.

- در مورد **دانایی در مورد باورها**؟
- در مورد **دانایی در مورد فرآیند استنتاج**؟

در محیط های چند عاملی،
استدلال یک عامل در مورد حالت های ذهنی سایر عامل ها مهم می شود.

رویداد ذهنی

Mental Event

فرآیندی که اشیای ذهنی را دست کاری می کند.

شیئی ذهنی

Mental Object

آنچه در سر (KB) عامل وجود دارد.

روابط بین **عامل ها و اشیای ذهنی**
believes, knows, wants, ...

Believes(Lois, Flies(Superman))

دانایی و باور

KNOWLEDGE AND BELIEF

Knowledge is justified true belief

دانایی = باور درست موجه

$Knows(a, p)$: agent a knows that proposition p is true.

KnowsWhether: می‌داند که آیا

$KnowsWhether(a, p) \Leftrightarrow Knows(a, p) \vee Knows(a, \neg p)$

$Knows(a, s) \Rightarrow Believes(a, s)$

KnowsWhat: می‌داند که

$KnowsWhat(a, \text{"PhoneNumber}(b)\text{"}) \Leftrightarrow$

$\exists x Knows(a, \text{"}x = \text{PhoneNumber}(b)\text{"}) \wedge x \in DigitStrings$

دانایی، زمان و کنش

KNOWLEDGE, TIME, AND ACTION

دانایی یک عامل (یا سایر عامل‌ها) در طول زمان تغییر می‌کند.

مدل‌سازی تأثیرات:

ترکیب استدلال در مورد رویدادها، کنش‌ها، زمان و دانایی

$T(\text{Believes}(\text{Lois}, \text{"Flies}(\text{Superman})\text{"}), \text{Today})$

$T(\text{Believes}(\text{Lois}, \text{"T(Flies(Superman), Yesterday)\text{"}), \text{Today}))$

۵

سیستم‌های
استدلال
برای
دسته‌ها

سیستم‌های استدلال برای دسته‌ها

نمونه‌ی سیستم‌های مخصوص استدلال برای دسته‌ها

منطق‌های توصیفی

Description Logics

- یک زبان صوری برای
تعریف ساخت و تعریف تعریف دسته‌ها
- دارای الگوریتم‌های کارآمد برای
تصمیم‌گیری در مورد روابط
زیرمجموعه / ابرمجموعه بین دسته‌ها

شبکه‌های معنایی

Semantic Networks

- یک نمایش گرافی از پایگاه دانایی
- دارای الگوریتم‌های کارآمد برای
استنتاج عضویت در دسته‌ها

شبکه‌های معنایی

SEMANTIC NETWORKS

شبکه‌های معنایی

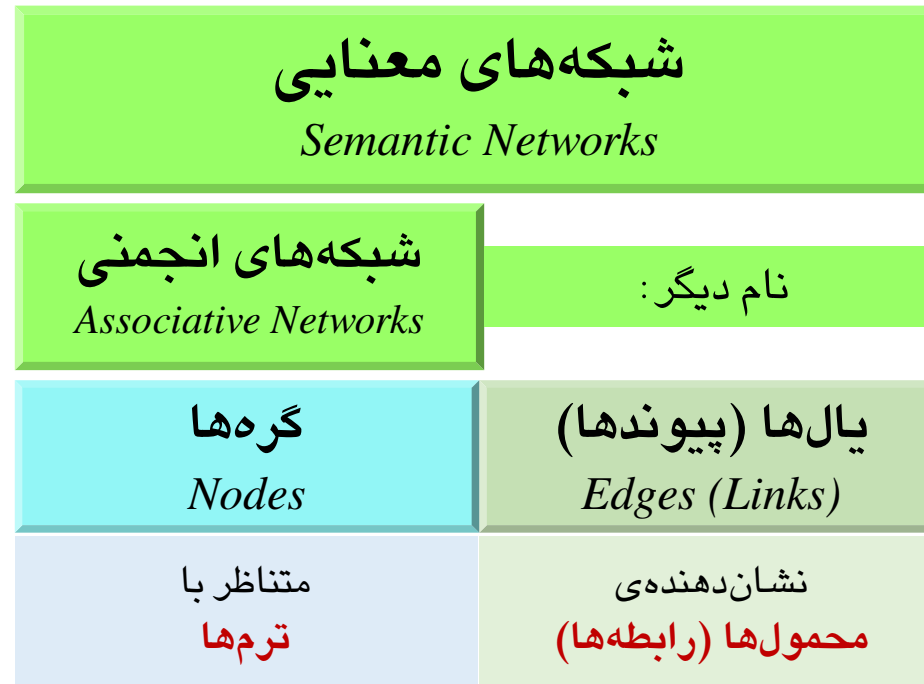
Semantic Networks

بازنمایی اشیاء، دسته‌های اشیاء و روابط آنها

امکان استفاده از مفهوم وراثت و استدلال وراثتی

شبکه‌های معنایی

SEMANTIC NETWORKS



شبکه‌های معنایی

نمایش یک صحنه با منطق مرتبه اول و شبکه‌های معنایی: مثال

SEMANTIC NETWORKS

$Inst(Block2, Block)$

$Color(Block2, Red)$

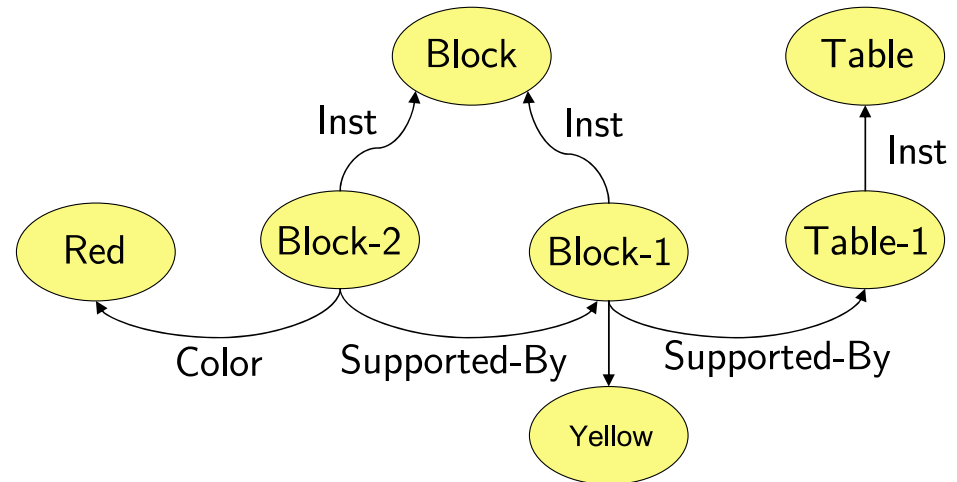
$SupportedBy(Block2, Block1)$

$Inst(Block1, Block)$

$Color(Block1, Yellow)$

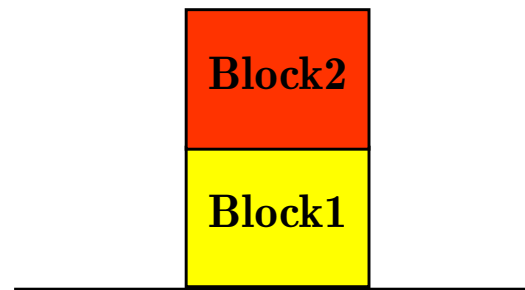
$SupportedBy(Block1, Table1)$

$Inst(Table1, Table)$



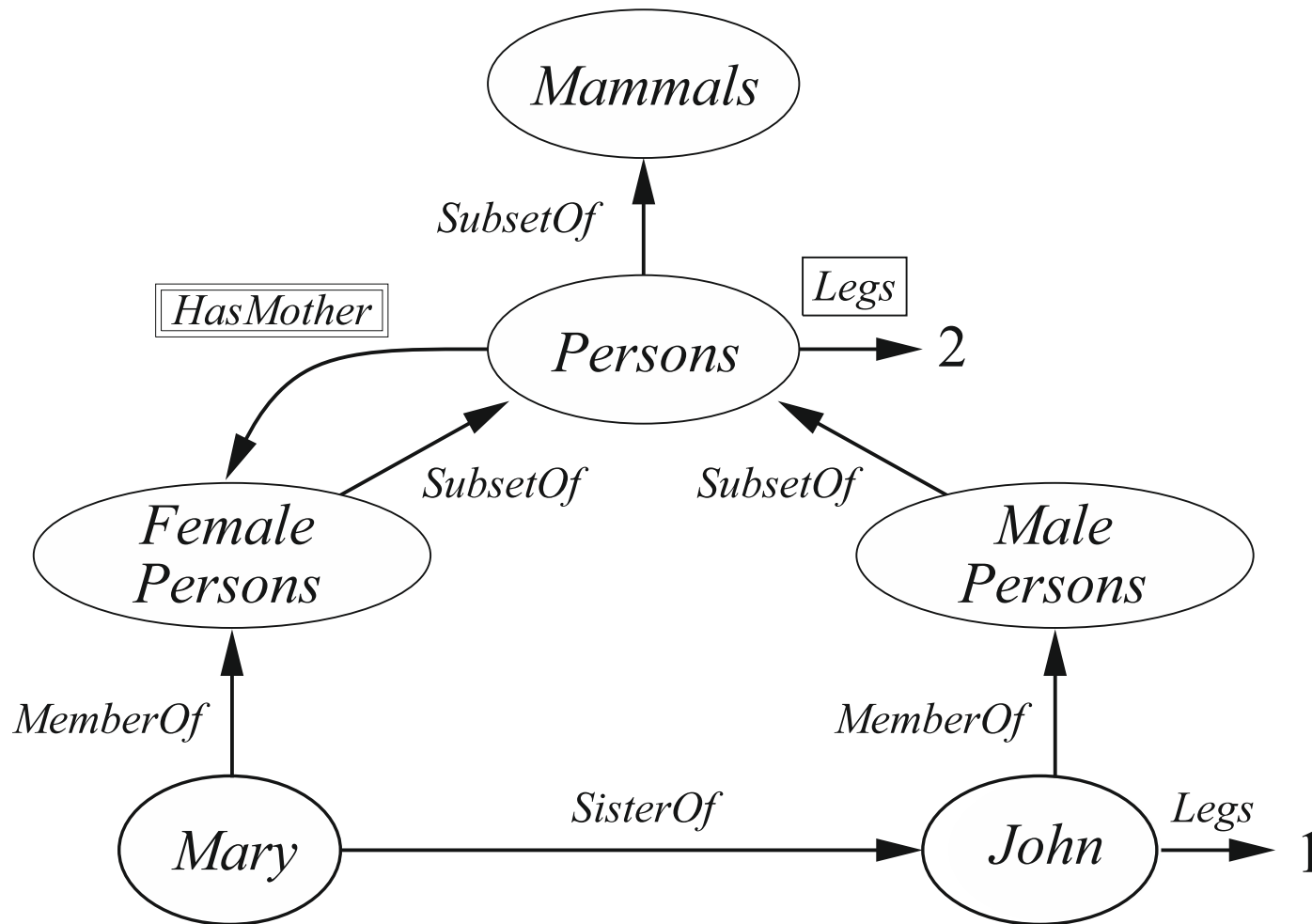
منطق مرتبه اول

شبکه‌ی معنایی



شبکه‌های معنایی

مثال

SEMANTIC NETWORKS

چهار شیء، چهار دسته و روابط آنها

شبکه‌های معنایی

نمادگذاری

SEMANTIC NETWORKS

Link Type	Semantic	Example
$A \xrightarrow{\text{Subset}} B$	$A \subseteq B$	$Cats \subseteq Mammals$
$A \xrightarrow{\text{Member}} B$	$A \in B$	$Tuna \in Cats$
$A \xrightarrow{R} B$	$R(A, B)$	$Tuna \xrightarrow{\text{Age}} 12$
$A \xrightarrow{\boxed{R}} B$	$\forall x \ x \in A \Rightarrow R(x, B)$	$Birds \xrightarrow{\boxed{\text{Legs}}} 2$
$A \xrightarrow{\boxed{\boxed{R}}} B$	$\forall x \ \exists y \ x \in A \Rightarrow y \in B \wedge R(x, y)$	$Birds \xrightarrow{\boxed{\boxed{\text{Parent}}}} Birds$

شبکه‌های معنایی

وراثت و سلسله‌مراتب IS-A

SEMANTIC NETWORKS

وراثت

Inheritance

نمونه‌ها، خصوصیات دسته‌ها را به ارث می‌برند.

رابطه‌ی وراثت در طول سلسله‌مراتب is-a منتشر می‌شود.

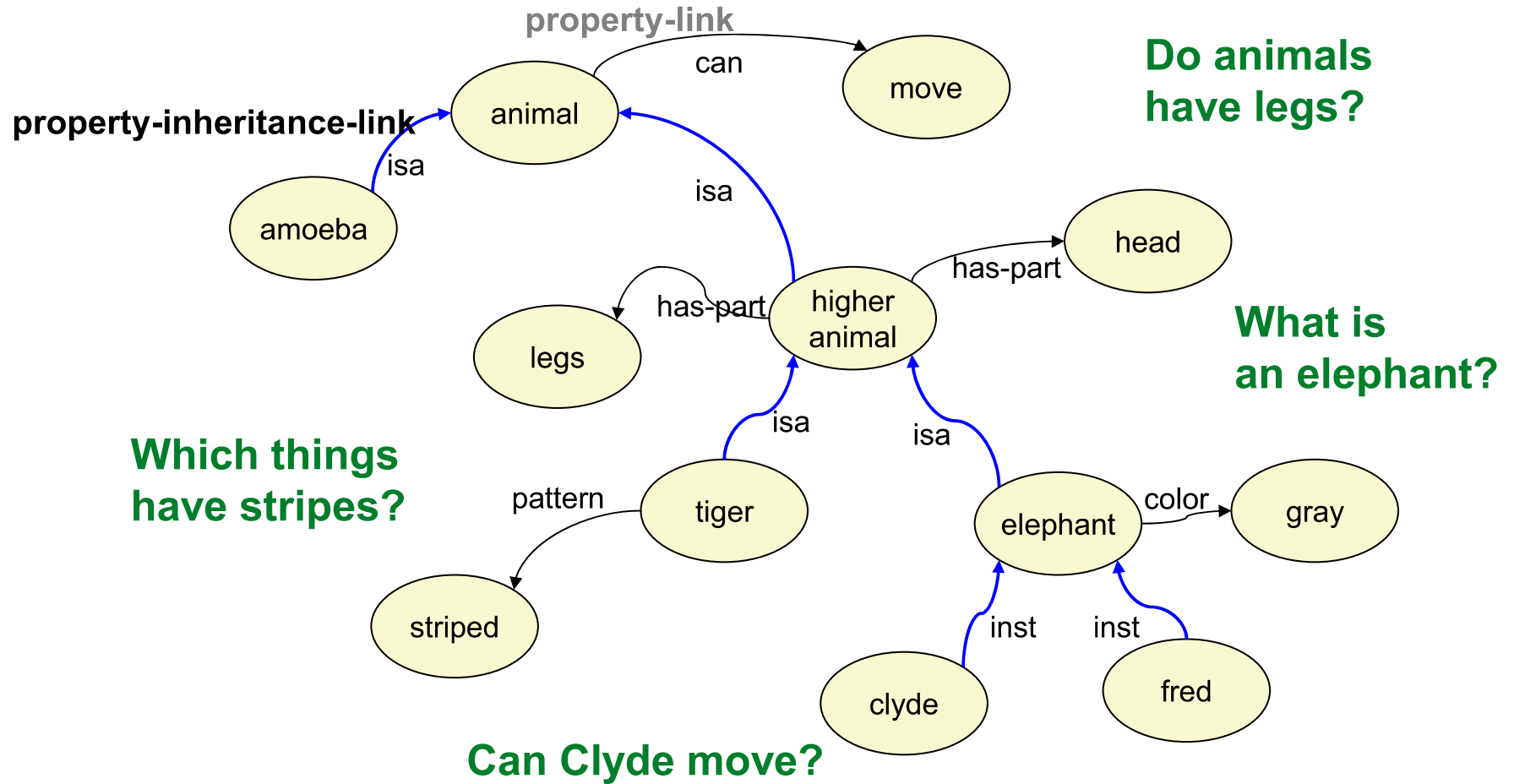
برای دسترسی به اطلاعاتی که مستقیماً اشاره نشده‌اند،
باید پیوندهای is-a و inst را دنبال کرد
تا مقادیر بر اساس وراثت تعیین شوند.

- **inst** $\in$ member of
- **is-a** $\subseteq$ subset of

شبکه‌های معنایی

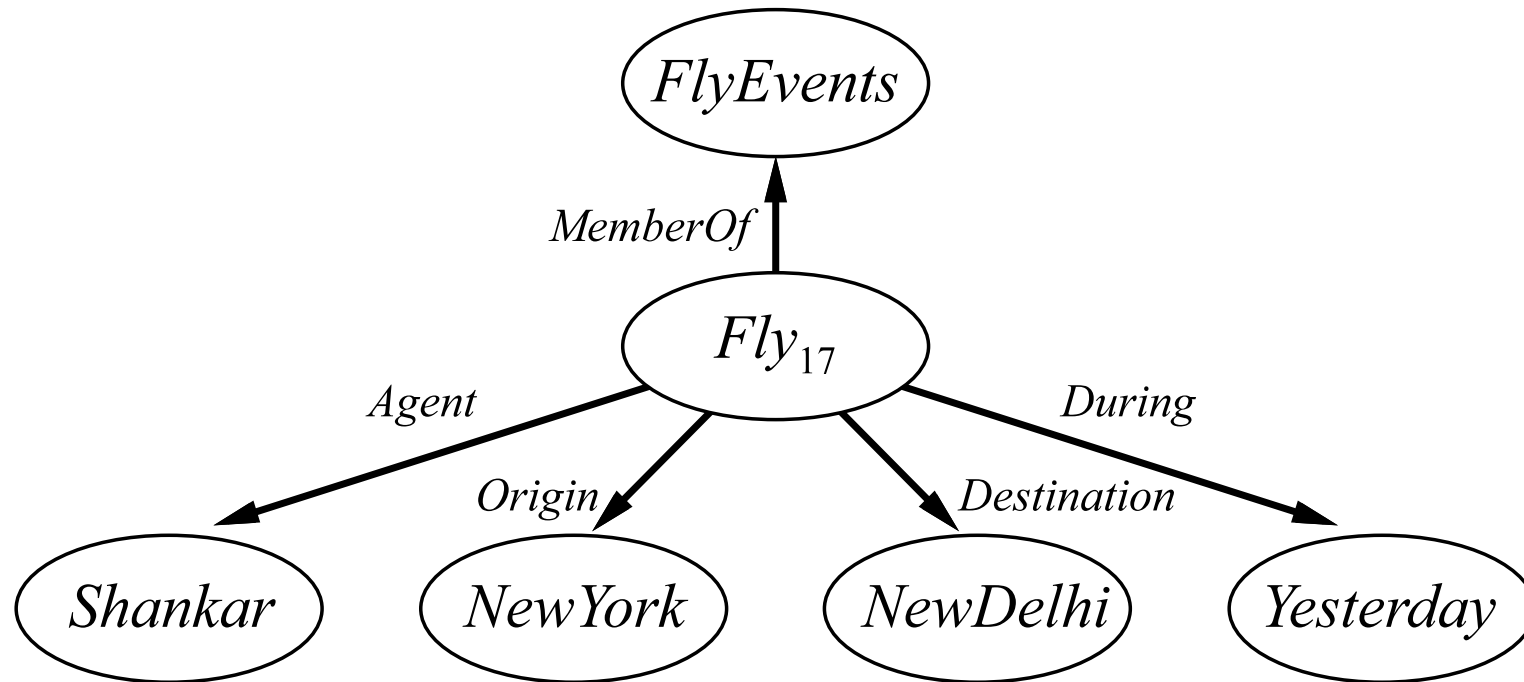
وراثت و سلسله‌مراتب IS-A: مثال

SEMANTIC NETWORKS



شبکه‌های معنایی

مثال

SEMANTIC NETWORKS

در شبکه‌های معنایی، پیوندها فقط می‌توانند رابطه‌های دودویی را بیان کنند.

راه‌حل: جسمیت بخشی به گزاره به عنوان یک رویداد

بازنمایی مقادیر پیش فرض: اعمال از طریق مکانیزم وراثت

قابها

FRAMES

قابها

Frames

بازنمایی دانایی در قالب کلیشه‌ها

Marvin Minsky (1975)

قابها

نمایش یک صحنه با منطق مرتبه اول و قابها: مثال

FRAMES

Inst(Block2, Block)

Color(Block2, Red)

SupportedBy(Block2, Block1)

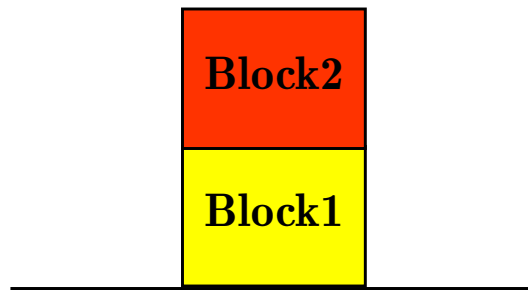
Inst(Block1, Block)

Color(Block1, Yellow)

SupportedBy(Block1, Table1)

Inst(Table1, Table)

منطق مرتبه اول



Frame	Attribute (slots)	Werte (fillers)
block-2 :	inst :	block
	color :	red
	supported-by :	block-1
	...	...

Frame	Attribute (slots)	Werte (fillers)
block-1 :	inst :	block
	color :	yellow
	supported-by :	table-1
	...	...

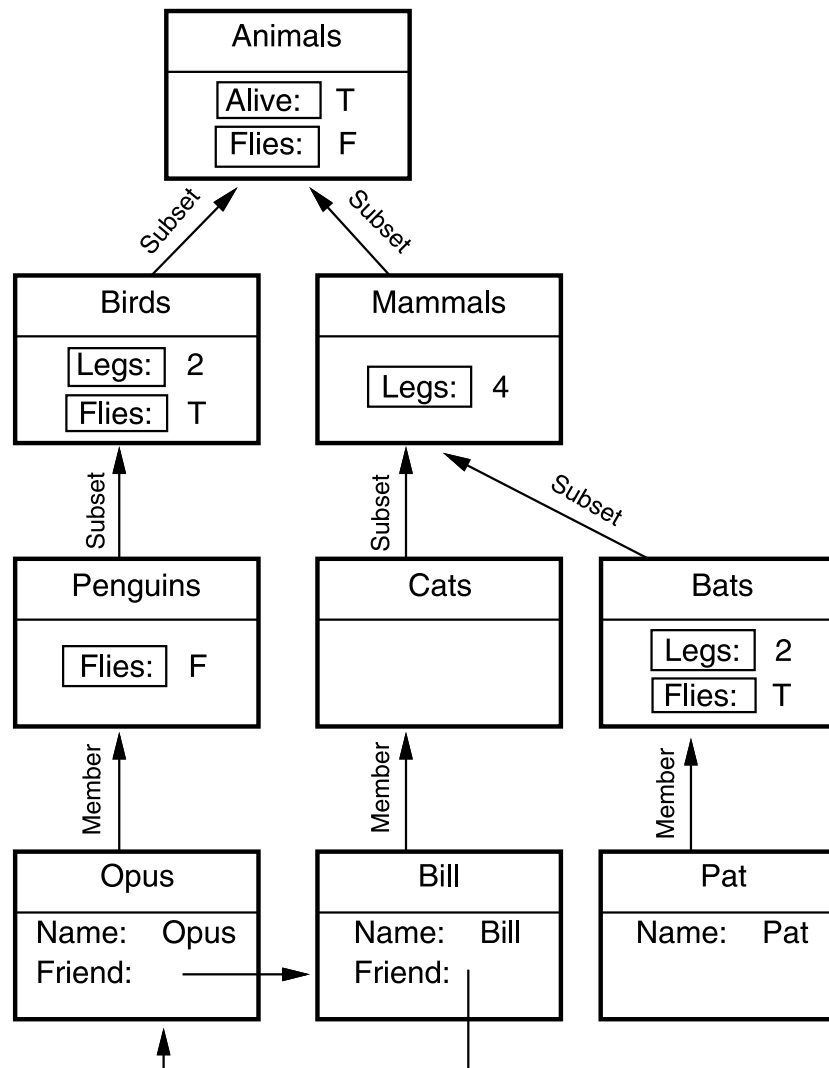
Frame	Attribute (slots)	Werte (fillers)
table-1 :	inst :	table
	color :	
	supported-by :	
	...	...

قابها

قابها

مقایسه‌ی «پایگاه دانایی مبتنی بر قابها» و «منطق مرتبه اول»

FRAME-BASED KB VS. FOL



(a) A frame-based knowledge base

Rel(Alive,Animals,T)

Rel(Flies,Animals,F)

Birds $\subset$ Animals

Mammals $\subset$ Animals

Rel(Flies,Birds,T)

Rel(Legs,Birds,2)

Rel(Legs,Mammals,4)

Penguins $\subset$ Birds

Cats $\subset$ Mammals

Bats $\subset$ Mammals

Rel(Flies,Penguins,F)

Rel(Legs,Bats,2)

Rel(Flies,Bats,T)

Opus $\in$ Penguins

Bill $\in$ Cats

Pat $\in$ Bats

Name(Opus,"Opus")

Name(Bill,"Bill")

Friend(Opus,Bill)

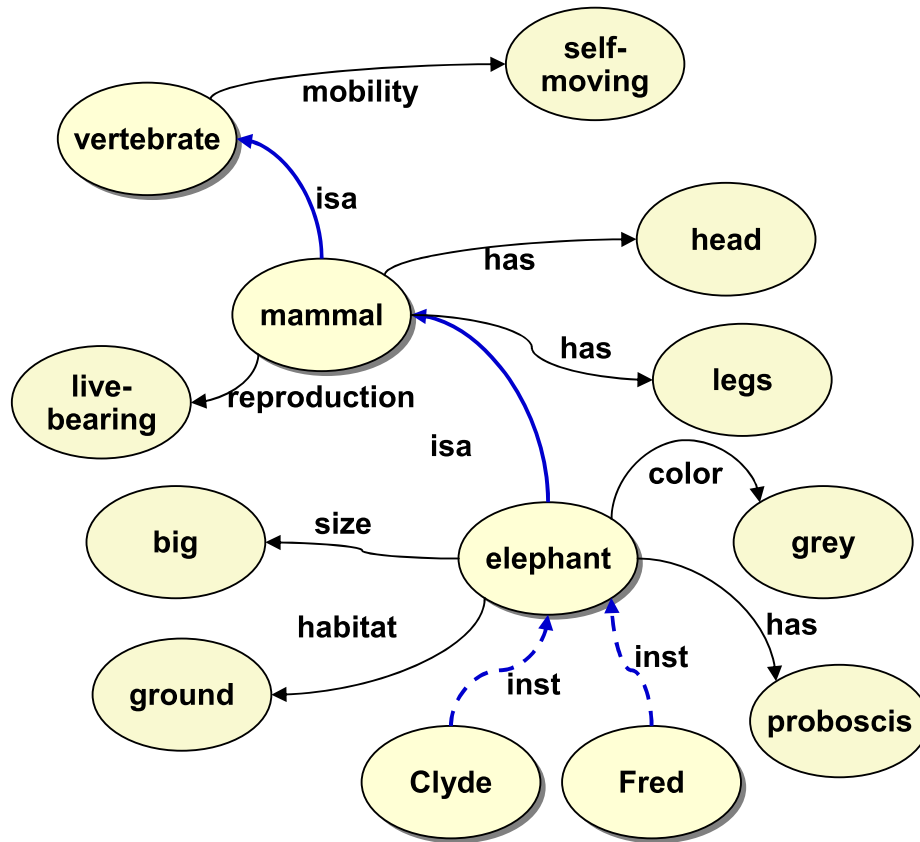
Friend(Bill,Opus)

Name(Pat,"Pat")

(b) Translation into first-order logic

وراثت

در شبکه‌های معنایی و قاب‌ها

INHERITANCE IN SEMANTIC NETS AND FRAMES

object	property	value
mammal :	isa :	vertebrate
	reproduction :	livebearing
	has :	head, legs
	...	...

object	property	value
elephant :	isa :	mammal
	color :	grey
	has :	proboscis
	size :	big
	habitat :	Boden

object	property	value
Clyde :	inst :	elephant
	color :	grey
	has :	proboscis
	size :	big
	habitat :	ground

منطق توصیفی

DESCRIPTION LOGIC

منطق توصیفی

Description Logic

منطق و نمادگذاری طراحی شده برای توصیف
تعریف‌ها و خصوصیات دسته‌ها

عملیات استنتاج تعریف شده برای منطق توصیفی

بررسی اینکه آیا یک دسته زیرمجموعه‌ی یک دسته‌ی دیگر است یا خیر
(به وسیله‌ی مقایسه‌ی تعریف آنها)

سابسومسیون
Subsumption

بررسی اینکه آیا یک شیء متعلق به یک دسته است یا خیر

طبقه‌بندی
Classification

بررسی اینکه آیا
ضابطه‌ی عضویت در یک دسته به لحاظ منطقی ارضا پذیر است یا خیر

سازگاری
Consistency

منطق توصیفی معمولاً فاقد نقیض و ترکیب فصلی است (برای تضمین زمان استنتاج غیرنمایی).

منطق توصیفی

مثال: زبان CLASSIC

DESCRIPTION LOGIC

$Concept \rightarrow \mathbf{Thing} \mid ConceptName$
 $\mid \mathbf{And}(Concept, \dots)$
 $\mid \mathbf{All}(RoleName, Concept)$
 $\mid \mathbf{AtLeast}(Integer, RoleName)$
 $\mid \mathbf{AtMost}(Integer, RoleName)$
 $\mid \mathbf{Fills}(RoleName, IndividualName, \dots)$
 $\mid \mathbf{SameAs}(Path, Path)$
 $\mid \mathbf{OneOf}(IndividualName, \dots)$
 $Path \rightarrow [RoleName, \dots]$

The syntax of descriptions in a subset of the CLASSIC language.

$Bachelor = And(Unmarried, Adult, Male) .$

The equivalent in first-order logic would be

$Bachelor(x) \Leftrightarrow Unmarried(x) \wedge Adult(x) \wedge Male(x) .$

منطق توصیفی

مثال: زبان CLASSIC

DESCRIPTION LOGIC

Notice that the description logic has an algebra of operations on predicates, which of course we can't do in first-order logic. Any description in CLASSIC can be translated into an equivalent first-order sentence, but some descriptions are more straightforward in CLASSIC. For example, to describe the set of men with at least three sons who are all unemployed and married to doctors, and at most two daughters who are all professors in physics or math departments, we would use

*And(Man, AtLeast(3, Son), AtMost(2, Daughter),
 All(Son, And(Unemployed, Married, All(Spouse, Doctor))),
 All(Daughter, And(Professor, Fills(Department, Physics, Math)))) .*

۶

استدلال با
اطلاعات
پیش فرض

استدلال با اطلاعات پیش فرض

REASONING WITH DEFAULT INFORMATION

"The following courses are offered: CS101, CS102, CS106, EE101"

How many courses are offered?

- Four (db)
 - Assume that this information is complete (not asserted ground atomic sentences are false)
= CLOSED WORLD ASSUMPTION (CWA)
 - Assume that distinct names refer to distinct objects
= UNIQUE NAMES ASSUMPTION (UNA)
- Between one and infinity (logic) = OPEN WORLD ASSUMPTION (OWA)
 - Does not make these assumptions (CWA, UNA)
 - Requires completion.

TRUTH MAINTENANCE SYSTEMS (TMS)

Many of the inferences have default status rather than being absolutely certain

- Inferred facts can be wrong and need to be retracted
= **BELIEF REVISION**.
- Assume KB contains sentence P and we want to execute $\text{TELL}(KB, \neg P)$
 - To avoid contradiction: $\text{RETRACT}(KB, P)$
 - But what about sentences inferred from P ?

Truth maintenance systems are designed to handle these complications.



مثال :
دنیای
خرید
اینترنتی

دنیای خرید اینترنتی

THE INTERNET SHOPPING WORLD

A Knowledge Engineering example

An agent that helps a buyer to find product offers on the internet.

IN = product description (*precise* or $\neg$ *precise*)

OUT = list of webpages that offer the product for sale.

Environment = WWW

Percepts = web pages (character strings)
Extracting useful information required.

THE INTERNET SHOPPING WORLD

Find relevant product offers

$$\begin{aligned} & \text{RelevantOffer}(\text{page}, \text{url}, \text{query}) \Leftrightarrow \\ & \text{Relevant}(\text{page}, \text{url}, \text{query}) \wedge \text{Offer}(\text{page}) \end{aligned}$$

- Write axioms to define $\text{Offer}(x)$
- Find relevant pages: $\text{Relevant}(x, y, z)$?
 - Start from an initial set of stores.
 - What is a relevant category?
 - What are relevant connected pages?
- Require rich category vocabulary.
 - Synonymy and ambiguity
- How to retrieve pages: $\text{GetPage}(\text{url})$?
 - Procedural attachment

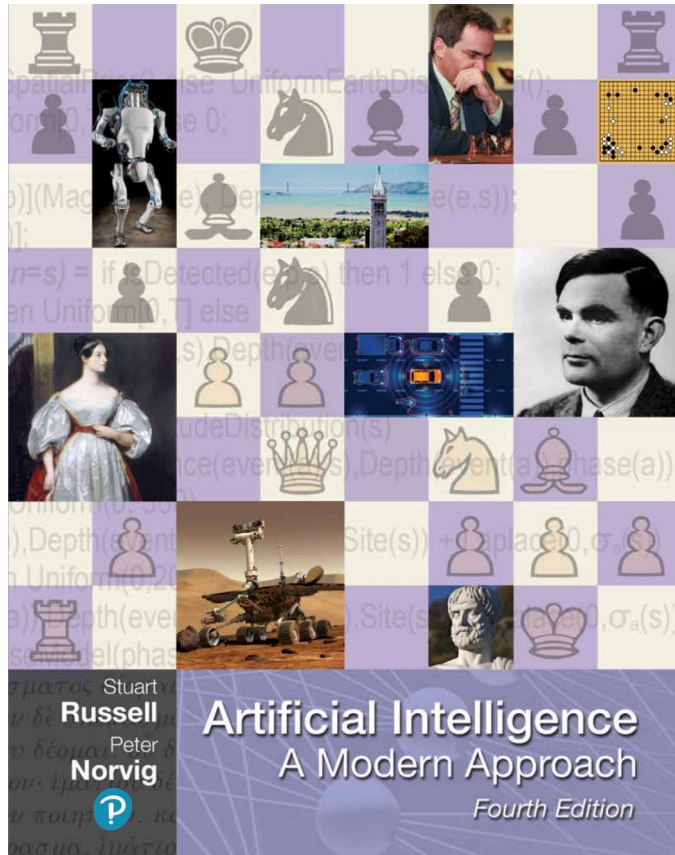
Compare offers (information extraction).

هوش مصنوعی

بازنمایی دانایی



منابع



Stuart Russell and Peter Norvig,
Artificial Intelligence: A Modern Approach,
 4th Edition, Prentice Hall, 2020.

Chapter 10

CHAPTER 10

KNOWLEDGE REPRESENTATION

In which we show how to represent diverse facts about the real world in a form that can be used to reason and solve problems.

The previous chapters showed how an agent with a knowledge base can make inferences that enable it to act appropriately. In this chapter we address the question of what *content* to put into such an agent's knowledge base—how to represent facts about the world. We will use first-order logic as the representation language, but later chapters will introduce different representation formalisms such as hierarchical task networks for reasoning about plans (Chapter 11), Bayesian networks for reasoning with uncertainty (Chapter 13), Markov models for reasoning over time (Chapter 17), and deep neural networks for reasoning about images, sounds, and other data (Chapter 21). But no matter what representation you use, the facts about the world still need to be handled, and this chapter gives you a feeling for the issues.

Section 10.1 introduces the idea of a general ontology, which organizes everything in the world into a hierarchy of categories. Section 10.2 covers the basic categories of objects, substances, and measures; Section 10.3 covers events; and Section 10.4 discusses knowledge about beliefs. We then return to consider the technology for reasoning with this content: Section 10.5 discusses reasoning systems designed for efficient inference with categories, and Section 10.6 discusses reasoning with default information.

10.1 Ontological Engineering

In “toy” domains, the choice of representation is not that important; many choices will work. Complex domains such as shopping on the Internet or driving a car in traffic require more general and flexible representations. This chapter shows how to create these representations, concentrating on general concepts—such as *Events*, *Time*, *Physical Objects*, and *Beliefs*—that occur in many different domains. Representing these abstract concepts is sometimes called **ontological engineering**.

We cannot hope to represent *everything* in the world, even a 1000-page textbook, but we will leave placeholders where new knowledge for any domain can fit in. For example, we will define what it means to be a physical object, and the details of different types of objects—robots, televisions, books, or whatever—can be filled in later. This is analogous to the way that designers of an object-oriented programming framework (such as the Java Swing graphical framework) define general concepts like *Window*, expecting users to use these to define more specific concepts like *SpreadsheetWindow*. The general framework of concepts is called an **upper ontology** because of the convention of drawing graphs with the general concepts at the top and the more specific concepts below them, as in Figure 10.1.

Ontological
engineering

Upper ontology