

Логика описания и ОНТОЛОГИИ

Курс: Концептуальное
моделирование предметных
областей





План лекции

- Логика описания
- Базовая логика описания ALC
- Расширения логики описания ALC
- Табличный вывод в логиках описания
- Языки онтологий OWL и OWL2
- Системы логического вывода



Встречающиеся переводы термина

- **Описательные логики**
- Логики описаний
- Дескрипционные логики
- Дескриптивные логики
- Description logics
- DL



Синтаксис базовой логики описаний ALC

Множество концептов логики ALC задается индуктивным определением:

- Символы \perp и \top – концепты (ложь и истина, пустой и полный концепты)
- Всякий атомарный концепт A является концептом
- Если C – концепт, то $\neg C$ – концепт (дополнение)
- Если C и D – концепты, то $C \sqcap D$, $C \sqcup D$ – концепты (пересечение и объединение)
- Если C – концепт, а R – атомарная роль, то $\exists R.C$, $\forall R.C$ — концепты
- никакие другие выражения концептами не являются



Семантика логики описаний ALC

- Интерпретация – пара $I = (\Delta, f^I)$, состоящая из непустого множества Δ , называемого областью интерпретации, и интерпретирующей функции f^I , которая сопоставляет:
 - каждому атомарному концепту $A \in \text{CN}$ – произвольное подмножество $A^I \subseteq \Delta$
 - каждой атомарной роли $R \in \text{RN}$ – произвольное подмножество $R^I \subseteq \Delta \times \Delta$.
- Пример
 - Область интерпретации – множество всех людей.
 - M – атомарный концепт, обозначающих людей мужского пола
 - R - роль «быть родителем для»
 - Тогда концепт $\forall R.M$ интерпретируется как множество людей, у которых все дети мужского пола.



Логика ALC и логика предикатов

- Переход к логике предикатов
 - \top и \perp сопоставляются с формулами \top и \perp соответственно
 - атомарному концепту A_i сопоставляется формула $P_i(x)$
 - концептам $\neg C$, $C \sqcap D$, $C \sqcup D$ сопоставляются формулы $\neg \Phi_C(x)$, $\Phi_C(x) \wedge \Phi_D(x)$, $\Phi_C(x) \vee \Phi_D(x)$ соответственно
 - концепту $\exists R.C$ сопоставляется формула $\exists y (S_j(x,y) \wedge \Phi'_C(y))$
 - концепту $\forall R.C$ сопоставляется формула $\forall y (S_j(x,y) \rightarrow \Phi'_C(y))$,
где $\Phi'_C(y)$ получена из $\Phi_C(x)$ заменой всех вхождений x на y , а y на x
- Не все формулы логики предикатов могут быть получены из ALC



Теоретико-множественная интерпретация

- $(\perp)^I : \emptyset$ (пустое множество)
- $(\top)^I : \Delta$ (множество всех возможных объектов)
- $(C)^I : C \subseteq \Delta$ (множество объектов)
- $(\mathcal{R})^I : R \subseteq \Delta \times \Delta$ (бинарное отношение)
- $(C_1 \sqcap C_2)^I : C_1 \cap C_2$ (пересечение множеств)
- $(C_1 \sqcup C_2)^I : C_1 \cup C_2$ (объединение множеств)
- $(\neg C)^I := \Delta \setminus C$ (все объекты, не входящие в множество C)
- $(\forall \mathcal{R}. C)^I : \{p \mid \forall q . \langle p, q \rangle \in R \rightarrow q \in C\}$ (все значения роли R принадлежат множеству C)
- $(\exists \mathcal{R}. C)^I : \{p \mid \exists q . \langle p, q \rangle \in R \wedge q \in C\}$ (существует хотя бы одно значение отношения R , принадлежащее C)
- $(\leq_n \mathcal{R}. C)^I : \{p \mid \#\{q \mid \langle p, q \rangle \in R\} \geq n\}$ (множество значений отношения R , принадлежащих C , не меньше n)
- $(C_1 \sqsubseteq C_2)^I : C_1 \subseteq C_2$ (C_1 является подмножеством C_2)
- $(i:C)^I : I \in C$ (объект I принадлежит множеству C)



Терминология и факты

- Терминология (TBox) – конечный набор аксиом.
 - Терминологическая аксиома - выражение вида $C \sqsubseteq D$ или $C \equiv D$, C и D — концепты.
- Система фактов (ABox) – конечный набор утверждений s : C или $a R b$, где $a, b, c \in IN$ - индивиды, C – концепт, R – роль.
- Семантика ABox: интерпретация $I = (\Delta, f^I)$ расширяется индивидами:
 - каждому индивиду $a \in IN$ сопоставляется элемент области интерпретации $a^I \in \Delta$.

- $Human \equiv Man \cup Woman$ Mary: Parent
- $Human \sqsubseteq \forall hasChild.Human$ Mary hasChild Anna
- $Parent \equiv Human \sqcap \exists hasChild.\top$ Anna: $\neg Man$



Расширения логики ALC

- F – функциональность ролей: концепты $\leq 1 R$
 - существует не более одного R-последователя
- N – ограничения множественности (кардинальности) ролей: концепты $\leq n R$
 - существует не более n R-последователей
- Q – качественные ограничения множественности ролей: концепты вида $(\leq n R.C)$
 - существует не более n R-последователей, входящих в концепт C
- I – обратные роли: роли вида R^{-}
 - если R есть роль, то R^{-} – роль - обращение бинарного отношения
- O – номиналы $\{a\}$
 - если a – имя индивида, то $\{a\}$ – концепт - одноэлементное множество
- H – иерархия ролей $R \sqsubseteq S$
 - в TBox допускаются аксиомы вложенности ролей
- S – транзитивные роли $\text{Tr}(R)$
 - в TBox допускаются аксиомы транзитивности
- R – составные аксиомы вложенности ролей, композиции ролей $R \bullet S$
 - в TBox допускаются роли $R \bullet S \sqsubseteq R$, $R \bullet S \sqsubseteq S$ с условием ацикличности
- <http://www.cs.man.ac.uk/~ezolin/dl/>
- Разрешимые логики (примеры)
 - Язык OWL-LITE – логика SHIF
 - Язык OWL-DL – логика SHOIN



Логические задачи

- (T0) Совместность терминологии T;
- (T1) Выполнимость концепта C в терминологии T
 - Непротиворечивость, satisfiability
- (T2) Вложение концептов $C \sqsubseteq D$ в терминологии T
 - Поглощение, subsumption
- (T3) классификация терминологии T:
 - для всех атомарных концептов $A, B \in CN$, встречающихся в T, проверить вложение $A \sqsubseteq B$ в T и вернуть таксономию – частично упорядоченное множество всех атомарных концептов относительно вложения.
- (K0) выполнимость базы знаний K
 - выполнимость $A \text{Box } A$ относительно $T \text{Box } T$;
- (K1) принадлежность индивида a концепту C относительно K
 - проверка того, что $K \models a : C$;
- (K2) выборка экземпляров класса:
 - для заданного концепта C найти $\{a \in IN \mid K \models a : C\}$;
- (K3) классификация индивида
 - найти все минимальные по вложению атомарные концепты, содержащие заданный индивид a относительно K (или найти все атомарные концепты A, такие что $K \models a : A$ и не существует атомарного концепта A' , такого что $K \models a : A'$, $T \models A' \sqsubseteq A$ и $T \not\models A \sqsubseteq A'$).



Табличный алгоритм

- Необходимо проверить выполнимость концепта C_0
 - Стартовый ABox содержит $A_0 = \{x_0 : C_0\}$,
 - Функция $SAT(A)$
 - { если в A есть x : \perp для некоторого x , то вернуть 0;
 - если в A есть x : A и x : $\neg A$ для некоторых x и A , то вернуть 0;
 - если к A не применимо ни одно из правил, то вернуть 1;
 - если применимо \sqcap , \exists или \forall -правило, то применить любое, получив A' , вернуть $SAT(A')$;
 - если применимо \sqcup -правило, то применить его, получив A' и A'' , и:
 - { если $SAT(A') = 1$, то вернуть 1;
 - если $SAT(A'') = 1$, то вернуть 1;
 - вернуть 0; }
- Правила создают новые ABox
- Механизм блокировки (против закливания алгоритма)
 - Метки $L(x) = \{C \mid x:C \in A\}$
 - Точка x блокирует точку y , если есть предок точки y (входящая вершина в графе) и $L(x) > L(y)$.



Табличный алгоритм (2)

| Правило | Условия применения | Действие |
|------------------------|---|--|
| Π -правило | если 0. точка x — активная 1. $x: (C \sqcap D) \in \mathcal{A}$ 2. $x: C \notin \mathcal{A}$ или $x: D \notin \mathcal{A}$ | то $\mathcal{A}' := \mathcal{A} \cup \{x: C, x: D\}$ |
| \sqcup -правило | если 0. точка x — активная 1. $x: (C \sqcup D) \in \mathcal{A}$ 2. $x: C \notin \mathcal{A}$ и $x: D \notin \mathcal{A}$ | то $\mathcal{A}' := \mathcal{A} \cup \{x: C\}$ $\mathcal{A}'' := \mathcal{A} \cup \{x: D\}$ |
| \exists -правило | если 0. точка x — активная 1. $x: \exists R. C \in \mathcal{A}$ 2. нет такого y , что $xRy \in \mathcal{A}$ и $y: C \in \mathcal{A}$ | то создать новую точку y и $\mathcal{A}' := \mathcal{A} \cup \{xRy, y: C\}$ |
| \forall -правило | если 0. точка x — активная 1. $x: \forall R. C \in \mathcal{A}$ 2. есть такой y , что $xRy \in \mathcal{A}$ и $y: C \notin \mathcal{A}$ | то $\mathcal{A}' := \mathcal{A} \cup \{y: C\}$ |
| \mathcal{T} -правило | если 0. точка x — активная 1. $x: E \notin \mathcal{A}$ | то $\mathcal{A}' := \mathcal{A} \cup \{x: E\}$ |



Использование единого алгоритма

- Приведение к вложению

C невыполним $\iff C \sqsubseteq \perp$;
 $C \equiv D \iff C \sqsubseteq D$ и $D \sqsubseteq C$;
 C и D не пересекаются $\iff C \sqcap D \sqsubseteq \perp$.

- Приведение к выполнимости

$C \sqsubseteq D \iff$ концепт $C \sqcap \neg D$ невыполним;
 $C \equiv D \iff$ концепты $C \sqcap \neg D$ и $D \sqcap \neg C$ оба невыполнимы;
 C и D не пересекаются \iff концепт $C \sqcap D$ невыполним.



Конструкции OWL DL

| OWL DL | DL |
|-----------------|--|
| Nothing | \perp |
| Thing | \top |
| Class | C |
| ObjectProperty | R |
| unionOf | $C_3 := C_1 \sqcup C_2$ |
| complementOf | $C_2 := \neg C_1$ |
| oneOf | $C := \{I_1\} \sqcup \{I_2\} \sqcup \dots$ |
| allValuesFrom | $C_2 := \forall R.C_1$ |
| someValuesFrom | $C_2 := \exists R.C_1$ |
| hasValue | $C := \exists R.\{I\}$ |
| minCardinality | $C := \geq_n R$ |
| maxCardinality | $C := \leq_n R$ |
| cardinality | $C := =_n R$ |
| subClassOf | $C_1 \sqsubseteq C_2$ |
| equivalentClass | $C_1 \equiv C_2 \equiv \dots$ |

| OWL DL | DL |
|-----------------------------|---|
| subPropertyOf | $R_1 \sqsubseteq R_2$ |
| domain | $\leq_1 R \sqsubseteq C$ |
| range | $\top \sqsubseteq \forall R.C$ |
| inverseOf | $R_1 \equiv R_2^-$ |
| equivalentProperty | $R_1 \equiv R_2 \equiv \dots$ |
| TransitiveProperty | $R^+ \sqsubseteq R$ |
| FunctionalProperty | $\top \sqsubseteq \leq_1 R$ |
| InverseFunctional Property | $\top \sqsubseteq \leq_1 R^-$ |
| SymmetricProperty | $R \equiv R^-$ |
| DatatypeProperty | A |
| type | $I: C$ |
| sameAs | $I_1 = I_2 = \dots$ |
| differentFrom, AllDifferent | $I_1 \neq I_2 \neq \dots$ |
| disjointWith | $C_1 \sqsubseteq \neg C_2$ $C_1 \sqcap C_2 \equiv \perp$ |