СӨЖ 4

SWISH ортасында орындау

1. Браузерде https://swish.swi-prolog.org/ мекенжайын ашыңыз. Prolog бағдарламасын енгізуге арналған жаңа редакторды таңдаңыз.
2. 6-бөлімдегі толық кодты редакторға көшіріңіз. Сұраныстарды кодқа қоспаңыз.
3. Сұраныс өрісіне eligible(aidana, ai201). енгізіп, Run батырмасын басыңыз. Күтілетін логикалық нәтиже true.
4. eligible(X, ai201). сұранысын орындаңыз. Шешімдер жиыны: X = aidana. Қосымша шешім сұратылса, басқа студент табылмайды.
5. eligible(daniyar, ai201). сұранысын орындаңыз. Бастапқы кодта нәтиже false. Бұл мақсаттың осы бағдарламада дәлелденбегенін білдіреді.
6. Фактілерге passed(daniyar, cs101). қосып, 4-қадамды қайталаңыз. Енді шешімдер жиыны екі студенттен тұрады.
7. seat(ai201). жолын уақытша seat(cs102). деп ауыстырып, жарамдылық сұранысын қайталаңыз. AI201 үшін толық дәлел табылмайды. Жолды соңында қалпына келтіріңіз. seat предикатының жалғыз анықтамасын түгел өшірмеңіз: әдепкі SWI-Prolog баптауында анықталмаған предикатты шақыру қате береді.

Егер SWISH ашылмаса, сол мәтінді UTF-8 кодтауымен enrollment.pl файлына сақтаңыз. SWI-Prolog консолінде consult('файлдың толық жолы/enrollment.pl'). арқылы жүктеңіз. Windows жолында / таңбасын қолдануға болады. Содан кейін сұраныстарды енгізіңіз. Консоль ?- таңбасын өзі көрсетеді.

Жиі кездесетін техникалық себептер: соңғы нүктені ұмыту; X орнына x жазу; сұранысты фактілердің арасына қосу; ai201 мен AI201 атауларын шатастыру. AI201 бас әріптен басталғандықтан айнымалы болып оқылады. Әуелі кодты бастапқы нұсқамен салыстырып тексеріңіз.

Барлық пререквизитті тексеру

Мына ықшам жалпы ереже бірнеше пререквизиті бар пәнге жарамсыз болуы мүмкін:

prolog

```
eligible(X, C) :- prerequisite(C, P), passed(X, P).
```

Бұл код кемінде бір сәйкес P тапса жеткілікті деп есептейді. Егер пәнге екі міндетті пререквизит қажет болса, бірін меңгеру барлық талапты орындауға тең емес. Біздің негізгі кодта осы қате туындамас үшін AI201 үшін жалғыз CS101 талабы айқын көрсетілген.

Барлық міндетті пәнді қамтитын бірінші ретті шарттың үлгісі төмендегідей. Ол өзі де тек жеткілікті шарт ретінде жазылған:

∀x∀c [∀p(Prerequisite(c,p)→Passed(x,p))→Eligible(x,c)]$\forall x \forall c [\forall p(Prerequisite(c,p) \rightarrow Passed(x,p)) \rightarrow Eligible(x,c)]$

Мұнда Prerequisite(c, p) «c пәніне p пәні алдын ала қажет» дегенді білдіреді. Ішкі ∀ барлық қажетті пәнді қамтиды. Пререквизиттер тізімі толық болмағанда, осы шартты деректерден қалай тексеретініміз қосымша келісімді талап етеді. Толық емес тізіммен нақты рұқсат беруді негіздеуге болмайды.

Кванторлар ретінің толық жазылуы

«Әр студент кемінде бір пәнді меңгерген» тұжырымы:

∀x(Student(x)→∃c(Course(c)∧Passed(x,c)))$\forall x(Student(x) \rightarrow \exists c(Course(c) \wedge Passed(x,c)))$

«Барлық студент меңгерген кемінде бір ортақ пән бар» тұжырымы:

∃c(Course(c)∧∀x(Student(x)→Passed(x,c)))$\exists c(Course(c) \wedge \forall x(Student(x) \rightarrow Passed(x,c)))$

Екі формуланың айырмасын Айдана тек CS101-ді, Данияр тек MATH101-ді меңгерген жағдаймен тексеріңіз. Осы екі студент және осы екі пәннен тұратын мысалда бірінші формула ақиқат, екіншісі жалған.

Б қосымшасы: Z3 арқылы қорытындыны тексеру

Z3 формулалардың бір мезгілде орындалатын моделін іздейді. Python интерфейсіндегі Bool бульдік айнымалы жасайды, Solver шешкішті құрады, add шектеулерді енгізеді, check нәтиже береді. Implies импликацияны, And конъюнкцияны, Not

терістеуді жазады. Қажет орта: Python 3 және z3-solver пакеті. Орнату пәрмені: python -m pip install z3-solver. Бұл әрекет дәріске дейін жасалады.

```
python
from z3 import Bool, Solver, And, Implies, Not

p = Bool('passed_cs101')
q = Bool('seat_ai201')
r = Bool('eligible_ai201')

solver = Solver()
solver.add(Implies(And(p, q), r))
solver.add(p, q)

print(solver.check())   # sat

solver.add(Not(r))
print(solver.check())   # unsat
```

Бірінші тексеру ереже мен екі фактінің бір-біріне қайшы еместігін көрсетеді: кемінде бір модель бар. Екінші тексеруде R-дің терістеуін қосамыз. Нәтиже unsat, өйткені P мен Q және ереже R-ді талап етеді. Демек, R бастапқы базадан туындайды. sat жауабының өзі кез келген сұралған тұжырым дәлелденді деген мағына бермейді.

Өздік тәжірибе: solver.add(p, q) жолын solver.add(p) деп ауыстырыңыз. Енді q туралы алғышарт жоқ. Екі тексерудің нәтижесі де sat болуы тиіс: q жалған, r жалған болатын модель ережеге қайшы келмейді. Бұл қарсы модель R-дің жаңа базадан міндетті түрде шықпайтынын көрсетеді. q-дың жалған болуы бұл жерде модельдегі мүмкін мән, нақты өмірде орын жоқтығы туралы дерек емес.

unknown нәтижесі жалпы жағдайда шешкіштің берілген есепке қорытынды бере алмағанын білдіреді. Осы шағын бульдік мысалдағы күтілетін нәтижелер sat және unsat. Z3-тің барлық теория мен квантор үшін аяқталуын немесе шешім табуын уәде етпейміз.

В қосымшасы: Терминдер сөздігі

Термин	Анықтама және қысқа мысал
Формализация	Табиғи тілдегі білімді нақты ережесі бар тілге көшіру. «Екеуі де керек» дегенді ∧ арқылы беру.
Синтаксис	Рұқсат етілген жазу құрылымы. Passed(aidana, cs101).
Семантика	Таңбалар мен формулалардың мағынасын анықтау. Passed студенттің пәнді меңгергенін білдіреді.
Факт	Базада ақиқат деп қабылданған нақты тұжырым. Student(aidana).
Ереже	Шарт орындалғанда шығатын қорытынды. A ∧ B → C.
Аксиома	Модельде бастапқы негіз ретінде қабылданатын тұжырым.
Пікір	Контексте ақиқат не жалған болатын хабарлы тұжырым. «Орын бар».
Предикат	Объект қасиеті немесе қатынасы. Passed(x, c).
Арлық	Предикат аргументтерінің саны. Passed үшін 2.
Термин	Объектіні білдіретін өрнек. aidana, x, advisor(aidana).
Квантор	Айнымалы бойынша жалпылау не бар болуды білдіретін белгі. ∀, ∃.

Интерпретация	Таңбаларға объектілер мен ақиқаттық шарттарды сәйкестендіру.
Модель	Берілген формулалардың бәрін ақиқат ететін интерпретация.
Логикалық салдар	Базаның барлық моделінде ақиқат тұжырым. KB ⊨ q.
Қанағаттандырылу	Формулалар жиынының кемінде бір моделінің болуы.
Унификация	Өрнектерді бірдей ететін алмастыруды табу. x = aidana.
Литерал және клауза	Атом не оның терістеуі; литералдар дизъюнкциясы. P, ¬P; ¬P ∨ R.
Хорн клауза	Ең көбі бір оң литералы бар клауза. ¬P ∨ ¬Q ∨ R.

Г қосымшасы: Бақылау сұрақтары және жауап бағдарлары

1. **Пікір мен предикаттың айырмасы қандай?** Пікір тұтас тұжырым ретінде алынады. Предикат объектінің қасиеті не қатынасын аргументтер арқылы береді.
2. **P → Q қай жағдайда жалған?** P ақиқат, Q жалған болғанда ғана.
3. **P ∨ Q екі пікірдің қатар ақиқат болуына жол береді ме?** Иә, бұл қамтушы дизъюнкция.
4. **∀x(Student(x) → HasID(x)) қалай оқылады?** Әрбір студенттің идентификаторы бар.
5. **¬∀x P(x) неге тең?** ∃x ¬P(x), яғни P орындалмайтын кемінде бір объект бар.
6. **Унификация не үшін керек?** Ереже мен фактіге ортақ үйлесімді алмастыру табу үшін.
7. **KB ⊨ q мен бір модельде q ақиқат болуының айырмасы қандай?** Біріншісі KB-ның барлық моделін қамтиды, екіншісі тек бір мүмкін жағдайды сипаттайды.
8. **Prolog-та** false **қандай сақтықпен түсіндіріледі?** Осы аяқталатын іздеуде сұранысқа дәлел табылмады. Бұл нақты өмірдегі терістеудің автоматты дәлелі емес.
9. **Қорытындыны Z3 арқылы қалай тексеруге болады?** KB ∧ ¬q енгіземіз. unsat болса, KB ⊨ q. Алдымен базаның өзінің қайшылықсыздығын тексерген дұрыс.
10. **Логикалық дұрыстық неге деректердің шынайылығына кепіл болмайды?** Қорытындылау берілген алғышарттарды пайдаланады, олардың нақты әлемге сәйкестігін бөлек тексереміз.

Өздік тапсырма: кітапхана кітабын алуға рұқсат беру ережесін жазыңыз. Тұрақтылар мен предикаттардың сөздігін беріңіз. Бір жарамды, бір шарты орындалмаған және бір мәліметі жетіспейтін жағдай құрыңыз. Әр нәтиженің қандай алғышартқа сүйенетінін түсіндіріңіз. Тексеру өлшемдері: терминдер бірмағыналы, формула сөйлемге сәйкес, импликация бағыты дұрыс, белгісіздік пен терістеу ажыратылған.

Дереккөздер және қосымша оқу

1. UC Berkeley CS 188. Logic және The Language of Logic. Формалды тіл, модель және логикалық салдар туралы оқу материалы. https://inst.eecs.berkeley.edu/~cs188/textbook/logic/
2. UC Berkeley CS 188. First Order Logical Inference. Алмастыру және жалпыланған Modus Ponens. https://inst.eecs.berkeley.edu/~cs188/textbook/logic/first-order-logical-inference.html
3. Poole D., Mackworth A. Artificial Intelligence: Foundations of Computational Agents. 3rd edition. Propositional Definite Clauses. https://www.cs.ubc.ca/~poole/aibook/3e/html/ArtInt3e.Ch5.S3.html
4. SWI-Prolog. Getting started quickly. Бағдарламаны жүктеу және сұраныс орындау. https://www.swi-prolog.org/pldoc/man?section=quickstart
5. SWI-Prolog. (+)/1. Дәлелдеудің сәтсіздігі арқылы терістеу. https://www.swi-prolog.org/pldoc/man?predicate=%5C%2B%2F1
6. Microsoft. Online Z3 Guide. Z3 API in Python. https://microsoft.github.io/z3guide/programming/Z3%20Python/Introduction/
7. Soufflé. Tutorial. Datalog және оның кеңейтімдері. https://souffle-lang.github.io/tutorial

8. SWISH. SWI-Prolog for SHaring. Браузердегі практикалық орта. https://swish.swi-prolog.org/

Теорияны тереңдету үшін: Russell S., Norvig P. Artificial Intelligence: A Modern Approach, 4th edition, Logical Agents және First Order Logic тараулары; Brachman R., Levesque H. Knowledge Representation and Reasoning.