

지식표현과 추론 – 2

관계기반 지식표현

이건명

충북대학교 산업인공지능학과

인공지능 : 튜링 테스트에서 딥러닝까지

학습 내용

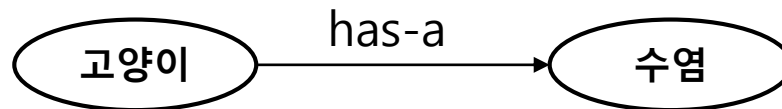
- 의미망, 스크립트, 온톨로지, 지식 그래프의 지식표현에 대해서 알아본다.
- 텍스트 처리 패키지를 사용하여 지식 그래프를 구성하는 방법을 알아본다.



1. 의미망

❖ 의미망(semantic network)

- 지식을 **이항 관계**(binary relation)의 집합으로 표현
- 노드와 방향성 간선으로 구성되는 **그래프**를 사용해 지식 표현
 - **노드** (node)
 - 대상, 개념, 행위, 상태, 사건
 - **간선** (edge)
 - 관계가 있는 노드 연결
 - 관계에 따른 **방향**
 - 관계의 의미를 나타내는 **라벨** 부여



의미망

❖ 의미망의 예

is-a(조류, 동물),
has-a(조류, 깃털),
can(조류, 날기),
cannot(펭귄, 날기),
do(동물, 호흡),
do(동물, 이동),
do(호랑이, 육식),
do(토끼, 초식),
can(펭귄, 수영),
do(포유류, 수유)

is-a(포유류, 동물)
can(조류, 알 낳기)
is-a(펭귄, 조류)
is-a(종달새, 조류)
has-a(동물, 피부)
has-a(호랑이, 날카로운 발톱)
is-a(토끼, 포유류)
is-a(트위티, 종달새)
is-a(바니, 토끼)

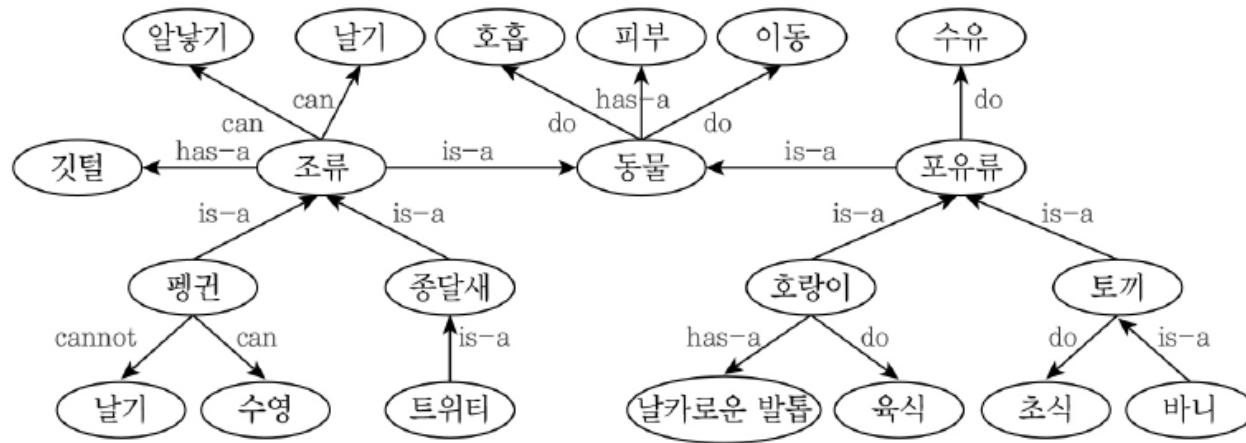


그림 3.7 의미망의 예



의미망

❖ 의미망에서 사용되는 **관계**(relationship)

▪ is-a

- **상위 클래스와 하위 클래스** 관계(예, is-a(조류,동물)), 또는 **클래스와 객체**의 관계(예, is-a(트위티,종달새))를 나타내어 **계층 관계** 표현
- 상위 계층의 속성을 **상속**
- **추이적**(transitive) **관계** 만족

$$\text{is-a}(\underline{X}, \underline{Y}) \wedge \text{is-a}(\underline{Y}, \underline{Z}) \rightarrow \text{is-a}(\underline{X}, \underline{Z}).$$

$$\text{is-a}(\underline{\text{펭귄}}, \underline{\text{조류}}) \wedge \text{is-a}(\underline{\text{조류}}, \underline{\text{동물}}) \rightarrow \text{is-a}(\underline{\text{펭귄}}, \underline{\text{동물}})$$

▪ has-a

- **전체-부분** 관계
- **part-of**와 역관계
 - **has-a**(X,Y)이면 **part-of**(Y,X) 성립
- **추이적 관계** 만족

$$\text{has-a}(\underline{X}, \underline{Y}) \wedge \text{has-a}(\underline{Y}, \underline{Z}) \rightarrow \text{has-a}(\underline{X}, \underline{Z}).$$

$$\text{part-of}(\underline{X}, \underline{Y}) \wedge \text{part-of}(\underline{Y}, \underline{Z}) \rightarrow \text{part-of}(\underline{X}, \underline{Z}).$$

의미망

❖ 다항 관계를 이항 관계로 전개하여 표현한 의미망

- 의미망은 이항 관계만을 표현
- 다항 관계는 “관계(relationship)”을 “객체(object)”로 간주하여 표현
 - 사물화(reification)
- 예. 길동이는 지난 가을부터 현재까지 고양이를 키우고 있다

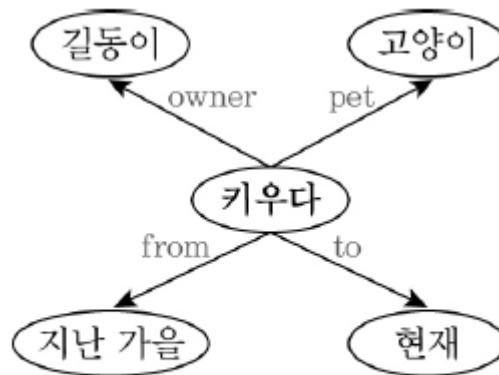
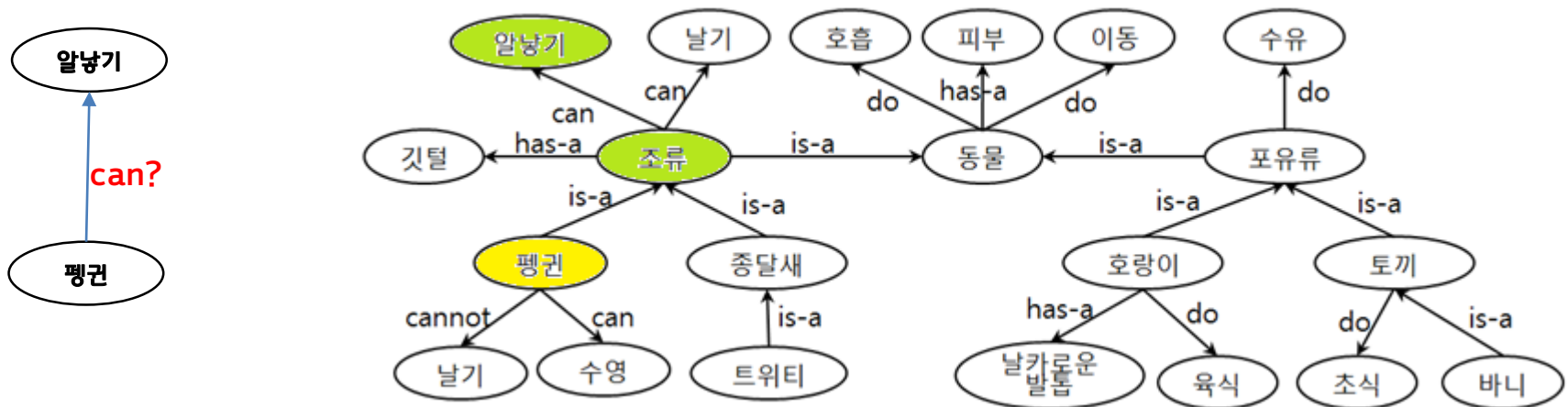


그림 3.8 다항 관계를 이항 관계로 전개하여 표현한 의미망

의미망의 추론

❖ 의미망에서 추론

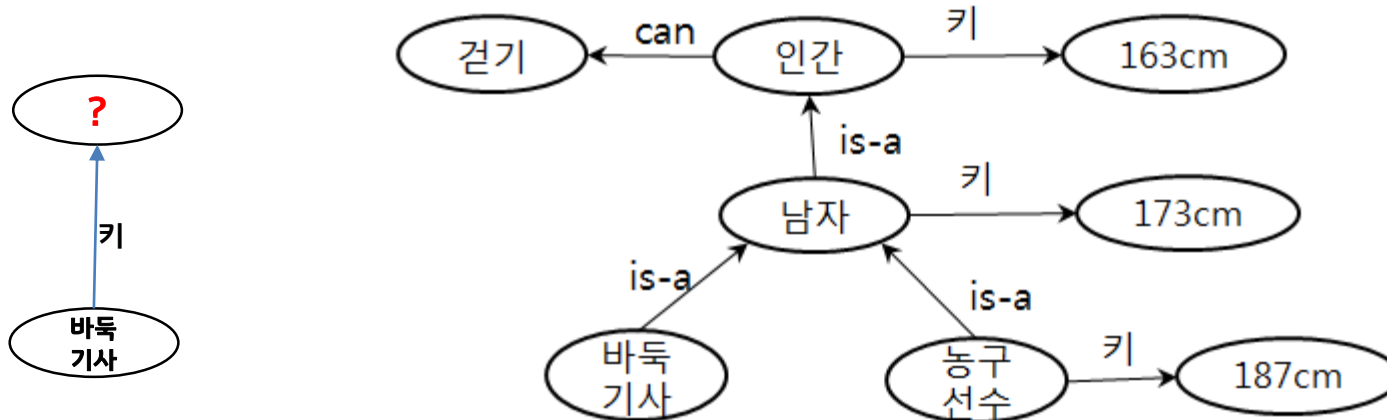
- 상속(inheritance) 이용
- 질문에 대한 의미망과 **지식**을 나타내는 의미망을 비교
- 질문 예. ‘**펭귄은 알을 낳는가?**’
 - can(펭귄,알낳기)에 해당
 - is-a 관계의 간선을 따라 조류 노드로 이동
 - can(조류, 알낳기)가 있으므로, 질문의 답은 참



의미망의 추론

❖ 의미망에서 추론

- **디폴트값**(default value, **기본값**)을 이용한 추론
- **상속 관계** 이용
- 예. 바둑 기사의 키는?



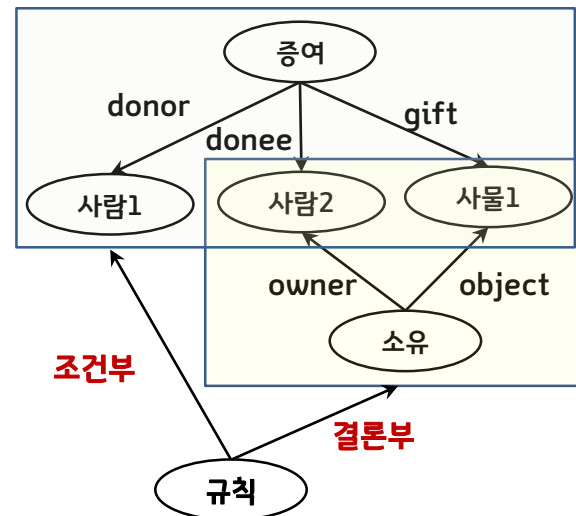
의미망의 추론

❖ 의미망에서 추론

- 주어진 지식으로부터 **새로운 사실**을 이끌어내는 **추론**도 가능
- 규칙의 의미망 표현
 - 예. **사람1**이 **사람2**에게 **사물1**을 주면, **사람2**는 **사물1**를 소유하게 된다

IF <사람1> gives <사물1> to <사람2>

Then <사람2> owns <사물1>

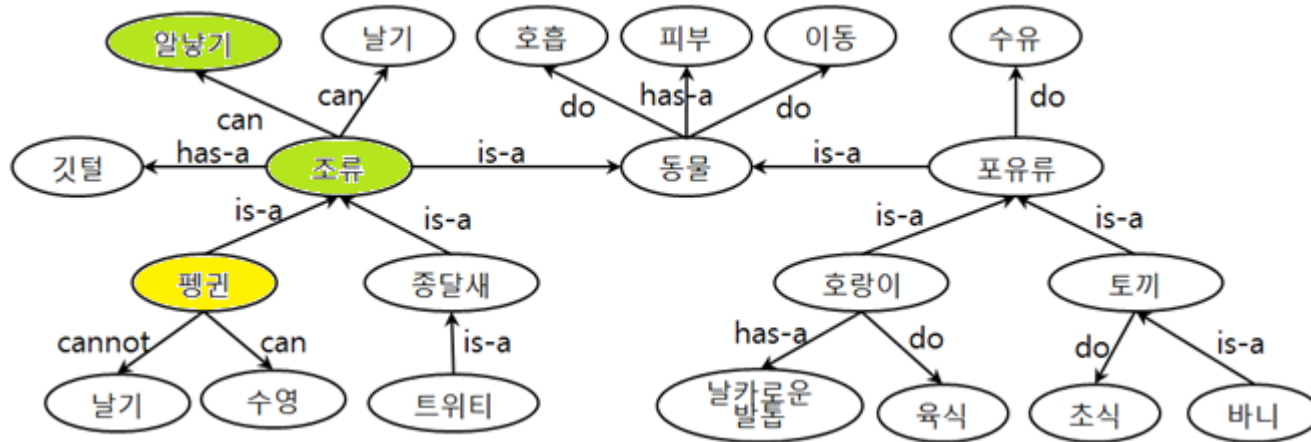


- 주어진 사실: **철수**가 **영희**에게 **반지**를 주었다
 - » 사람1=철수, 사람2=영희, 사물1=반지
 - » 추론 결과 : **영희**가 **반지**를 가지고 있다

의미망과 프레임

❖ 의미망의 프레임으로의 변환

- 노드별로 프레임 생성
- 노드에서 나가는 간선들을 슬롯(slot)으로 구성



frame-name	펭귄
frame-type	Class
is-a	조류
can	수영
cannot	날기

frame-name	조류
frame-type	Class
is-a	동물
can	날기 알날기
has-a	깃털

의미망

❖ 의미망 표현의 장점

- 지식을 **시각적**으로 표현할 수 있어, 직관적 이해 용이
- 노드 추가 또는 변경으로 비교적 **쉽게 지식의 추가 및 변경** 가능
- 개념의 **계층관계**를 정의하여 속성의 상속 관계 지정 가능
- 복잡한 지식을 **구조화**하여 표현 가능

❖ 의미망 표현의 단점

- 지식의 양이 많아지면 관리 **복잡**
- 개념이나 관계를 임의로 정의하기 때문에 **통일성 부족**
- **공유나 재사용**에 대한 **고려 없음**
- 논리적 **결합** 관계나 **인과** 관계를 기술하려고 하면 and, or, implies와 같은 링크 도입 필요
 - 일관성을 떨어뜨리고 추론과정을 복잡
- 기본적으로 **정적인 지식**의 표현
 - 추론 과정에서 동적으로 지식의 내용을 바꾸려면 그래프를 동적으로 바꿀 수 있도록 해야 함

2. 스크립트

❖ 스크립트(script)

- 일반적으로 발생할 수 있는 **전형적인 상황**에 대한 **절차적 지식**을 일목요연하게 표현
- 전형적인 상황에서 일어나는 **일련의 사건(event)**을 **시간적 순서**를 고려하여 기술하는 프레임과 같은 구조의 지식 표현
- 구성요소
 - **진입 조건(entry conditions)**
 - 스크립트에 기술된 사건들이 일어나기 전에 만족되어야 하는 전제조건
 - **역할자(roles)**
 - 스크립트에 관련된 사람 및 대상
 - **자산(properties)**
 - 사건 진행에 과정에서 사용되는 객체
 - **트랙(track)**
 - 어떤 스크립트에서 발생할 수 있는 일련의 사건들이 변형된 형태 식별자
 - **장면(scenes)**
 - 실제 일어나는 일련의 사건
 - **결과 조건(results)**
 - 스크립트의 장면에 있는 사건들이 일어난 이후에 만족되는 조건

스크립트 이름	식당
트랙(track)	패스트푸드 식당
역할자(roles)	고객, 점원
자산(properties)	카운터, 쟁반, 음식, 돈, 냅킨, 소금/후추/시럽/빨대
진입조건 (entry conditions)	고객이 배가 고프다. 고객은 돈이 있다.
장면 (scenes)	장면 1: 입장 고객이 식당에 들어선다. 고객은 카운터에서 줄을 서서 기다린다. 고객은 벽에 있는 메뉴를 보고, 주문할 것을 결정한다.
	장면 2: 주문 고객이 점원에게 주문을 한다. 고객은 점원에게 음식값을 지불한다. 점원이 주문한 것을 쟁반에 올려 놓는다.
	장면 3: 식사 고객은 소금/후추/냅킨/빨대 등을 집는다. 고객은 쟁반을 받아두고 빈자리에 가서 앉는다. 고객은 음식을 빨리 먹는다.
	장면 3-1(선택적) : 들고 가기(take-out) 고객이 음식을 가지고 나간다.
	장면 4: 퇴장 고객이 식탁을 치운다. 고객이 출입문 옆 쓰레기통에 버린다. 고객이 식당을 나간다.
결과 조건 (results)	고객은 더 이상 배고프지 않다. 고객의 돈이 줄었다. 고객은 만족스럽다. (선택적) 고객은 만족스럽지 못하다. (선택적)

스크립트

❖ 패스트푸드 식당에서의 상황을 표현한 스크립트.

스크립트 이름	식당
트랙(track)	패스트푸드 식당
역할자(roles)	고객, 점원
자산(properties)	카운터, 쟁반, 음식, 돈, 냅킨, 소금/후추/시럽/빨대
진입조건 (entry conditions)	고객이 배가 고프다. 고객은 돈이 있다.
장면 (scenes)	장면 1: 입장 고객이 식당에 들어선다. 고객은 카운터에서 줄을 서서 기다린다. 고객은 벽에 있는 메뉴를 보고, 주문할 것을 결정한다.
	장면 2: 주문 고객이 점원에게 주문을 한다. 고객은 점원에게 음식값을 지불한다. 점원이 주문한 것을 쟁반에 올려 놓는다.
	장면 3: 식사 고객은 소금/후추/냅킨/빨대 등을 집는다. 고객은 쟁반을 받아들고 빈자리에 가서 앉는다. 고객은 음식을 빨리 먹는다.
	장면 3-1(선택적) : 들고 가기(take-out) 고객이 음식을 가지고 나간다.
	장면 4: 퇴장 고객이 식탁을 치운다. 고객이 출입문 옆 쓰레기통에 버린다. 고객이 식당을 나간다.
결과 조건 (results)	고객은 더 이상 배고프지 않다. 고객의 돈이 줄었다. 고객은 만족스럽다. (선택적) 고객은 만족스럽지 못하다. (선택적)

고객이 식당에 들어오면 맨 먼저
뭘 하는가?

누가 돈을 지불하는가?

식사 후 식당을 나가기 전에 뭘 하
는가?

3. 온톨로지

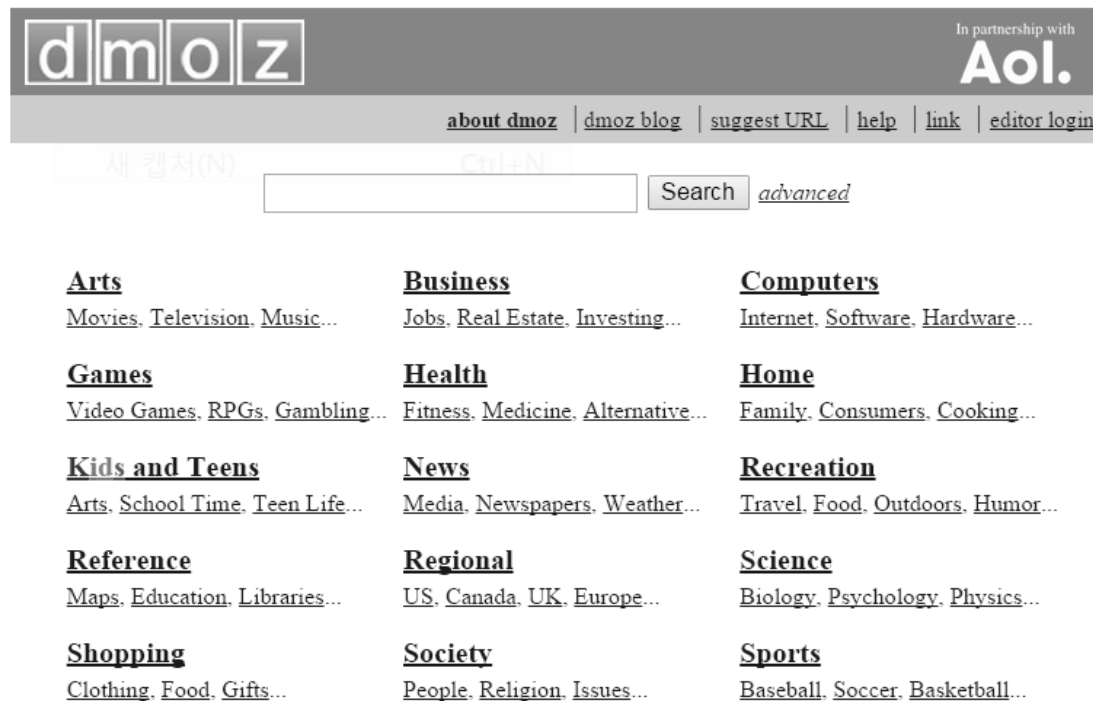
❖ 온톨로지(ontology)

- 철학에서 존재론(存在論)을 가리키는 말
- 어떤 영역(domain)의 지식을 공유하고 재사용할 수 있도록 해당 영역의 개념과 관계를 나타내는 어휘를 정의하고 이를 이용해 지식을 표현해 놓은 것
- 영역에 있는 개념, 개념에 대한 특성 및 속성, 이들 특성과 속성에 대한 제약조건, 영역에 있는 일부 개체에 대한 정보 기술(記述)
- 영역에 대한 공통된 어휘(vocabulary) 사용
- 영역에 대한 공통된 이해 지원
- 서로 간 토의를 통해 합의에 이른 것
- 컴퓨터에서 다룰 수 있는 형태로 정형화하여 표현

온톨로지

❖ 온톨로지의 사례

- Amazon.com의 온라인 쇼핑 카탈로그
- dmoz(Directory of Mozilla, www.dmoz.org)
 - 분야별로 정리한 디렉토리 서비스



온톨로지

❖ 온톨로지의 사례 – cont.

▪ UNSPSC(United Nations Standard Products and Services Code)

- 제품 및 서비스 용어

FAMILY CODE	DESCRIPTION
	COMMODITY CLASS DESCRIPTION
1010	LIVE ANIMALS
101015	LIVESTOCK
101016	BIRDS AND FOWL
101017	LIVE FISH
101018	SHELLFISH AND AQUATIC INVERTEBRATES
101019	INSECTS

▪ WordNet(wordnet.princeton.edu)

- 영어 단어의 어휘목록과 어휘목록 사이의 다양한 의미 관계를 기록

WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- [S:](#) (n) **ontology** ((computer science) a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities and their relations)
 - [domain category](#)
 - [S:](#) (n) **computer science, computing** (the branch of engineering science that studies (with the aid of computers) computable processes and structures)
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
- [S:](#) (n) **ontology** (the metaphysical study of the nature of being and existence)
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
 - [derivationally related form](#)
 - [W:](#) (adj) **ontological** [Related to: [ontology](#)] (of or relating to ontology) "ontological speculations"

온톨로지

❖ 온톨로지의 사례 – cont.

- **UMLS**(Unified Medical Language System)
 - 의료 영역의 여러 용어체계 총괄
- **SNOMED-CT**(Systematized Nomenclature of Medicine – Clinical Terms)
 - 의료용어, 유사어, 코드
- **LOINC**(Logical Observation Identifiers Names and Codes)
 - 병원 검사 명칭 및 임상 용어 표준 체계
- **GO**(Gene Ontology)
 - 유전자와 유전자 산물

온톨로지

❖ 온톨로지의 사례 – cont.

- 의료 용어 온톨로지 SNOMED-CT의 **감기**(common cold)에 대한 항목

Parent(s):

(Select a parent to make it the "Current Concept".)

[Viral upper respiratory tract infection \(disorder\)](#)

Current Concept:

[Common cold \(disorder\)](#)

Child(ren):

(N=0) (Select a child to make it the "Current Concept".)

Current Concept:

Fully Specified Name: [Common cold \(disorder\)](#)

ConceptId: 82272006

Defining Relationships:

Is a [Viral upper respiratory tract infection \(disorder\)](#)
Causative agent [Virus \(organism\)](#)
Finding site [Upper respiratory tract structure \(body structure\)](#)
Pathological process [Infectious process \(qualifier value\)](#)

This concept is primitive.

Qualifiers:

[View Qualifying Characteristics and Facts](#)

Descriptions (Synonyms):

Fully Specified Name: [Common cold \(disorder\)](#)

Preferred: [Common cold](#)
Synonym: [Acute coryza](#)
Synonym: [Acute nasal catarrh](#)
Synonym: [Acute rhinitis](#)
Synonym: [Infective rhinitis](#)
Synonym: [Acute nasopharyngitis](#)
Synonym: [Infective nasopharyngitis](#)
Synonym: [Head cold](#)
Synonym: [Acute infective rhinitis](#)
Synonym: [Cold](#)
Synonym: [Acute nasopharyngitis, NOS](#)
Synonym: [Infective nasopharyngitis, NOS](#)

온톨로지

❖ 온톨로지의 지식표현

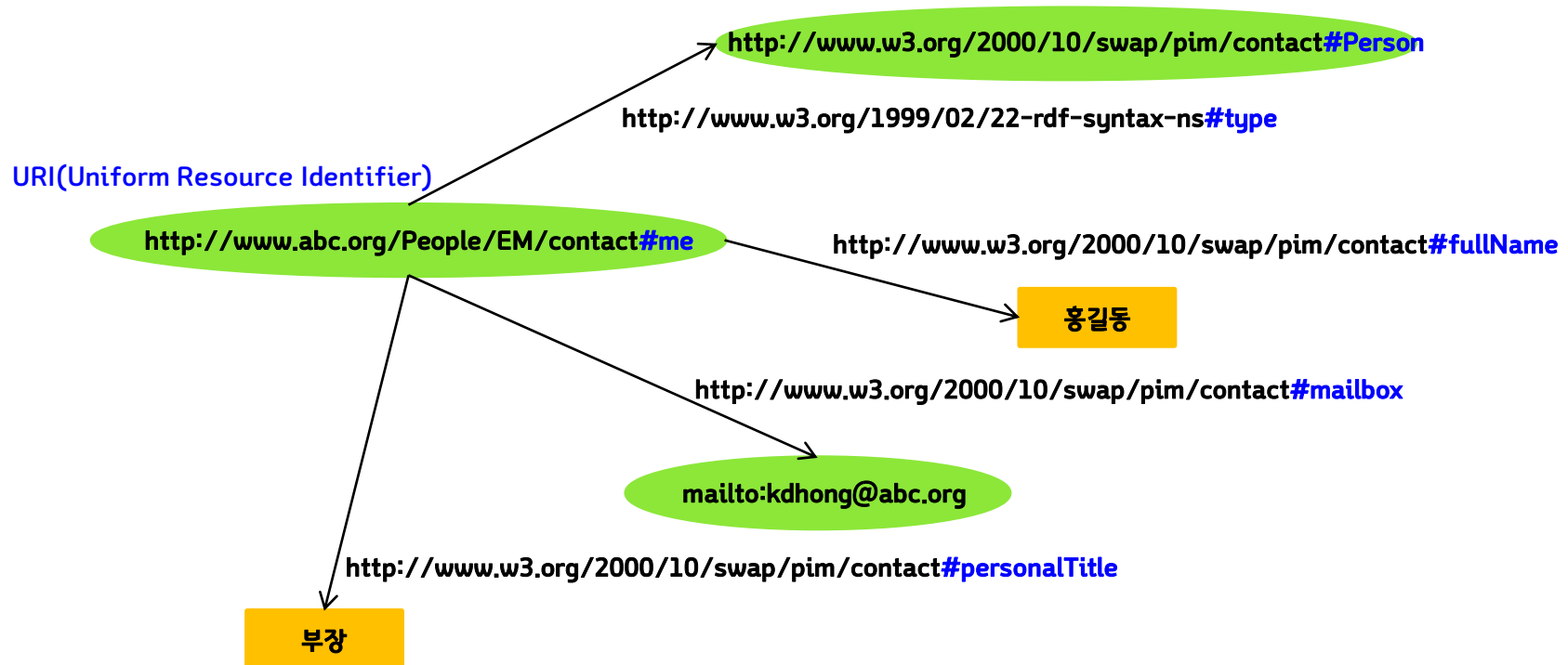
- 의미망과 비슷하게 **방향성**이 있는 **그래프**로 표현 가능
- **RDF**(Resource Description Framework)
 - 자원에 대한 **메타데이터**(metadata, 데이터에 대한 데이터)를 기술하는 명세 (specification)
 - 웹 상의 데이터를 상호 운용 가능한 형식으로 표현하고 공유할 수 있도록 설계
 - RDF에서 모든 것(개체, 관계, 개념 등)은 리소스로 간주
 - 각 리소스는 고유한 **URI** (Uniform Resource Identifier)로 식별
 - 사용자가 필요에 따라 새로운 용어나 관계를 정의할 수 있게 해주는 네임스페이스와 확장 가능한 어휘 구조 지원
 - 온톨로지 모델링, 시맨틱 웹, 데이터 통합, 지식 그래프, 개방형 데이터 등의 분야에서 사용

온톨로지

❖ 온톨로지의 지식표현

▪ RDF(Resource Description Framework)

- 특정 대상을 기술하는 간단한 문장(sentence)
 - ‘주어(subject)+서술어(predicate)+목적어(object)’로 파악
 - 자원(대상; resource) – 속성(특징, 주어와 목적어 관계, attribute) – 값(value)으로 분해
- abc 기관의 이메일이 kdhong@abc.org인 홍길동 부장이라는 사람



온톨로지

❖ 온톨로지의 지식표현

▪ RDF(Resource Description Framework) – Conf.

- RDF 데이터는 XML, N-Triples, Turtle, JSON-LD 등으로 표현 가능

• 튜플 형태의 표현

자원

속성

값

<http://www.abc.org/People/EM/contact#me>, <http://www.w3.org/2000/10/swap/pim/contact#fullName>, “홍길동”

<http://www.abc.org/People/EM/contact#me>, <http://www.w3.org/2000/10/swap/pim/contact#mailbox>,

<mailto:kdhong@abc.org>

<http://www.abc.org/People/EM/contact#me>, <http://www.w3.org/2000/10/swap/pim/contact#personalTitle>, “부장”

<http://www.abc.org/People/EM/contact#me>, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>,

<http://www.w3.org/2000/10/swap/pim/contact#Person>

• XML 형태의 표현

```
<?xml version="1.0"?>
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
```

```
  <contact:Person rdf:about="http://www.abc.org/People/EM/contact#me">
```

```
    <contact:fullName>홍길동</contact:fullName>
```

```
    <contact:mailbox rdf:resource="mailto:kdhong@abc.org"/>
```

```
    <contact:personalTitle>부장</contact:personalTitle>
```

```
  </contact:Person>
```

```
</rdf:RDF>
```

[rdf:RDF](#) 내부에 namespace 선언

[rdf:about](#) subject URI

[rdf:resource](#) object URI

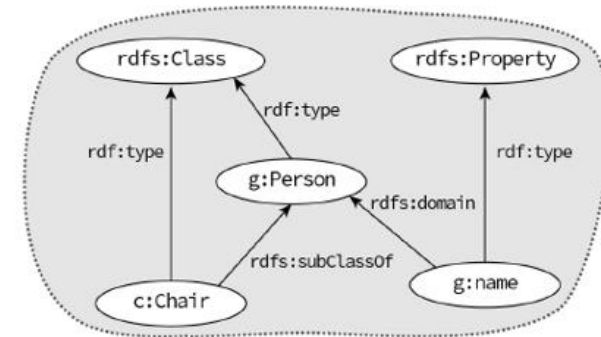
온톨로지

❖ 온톨로지의 지식표현 – cont.

▪ **RDFS**(Resource Description Framework Schema, RDF 스키마)

- RDF를 사용하여 온톨로지를 표현할 때 사용할 관련 어휘 온톨로지를 정의하는 언어
- 리소스 간의 관계와 클래스 구조를 기술하는 데 사용되는 언어
- 클래스, 속성, 계층 구조 등의 의미론적 관계 정의 가능

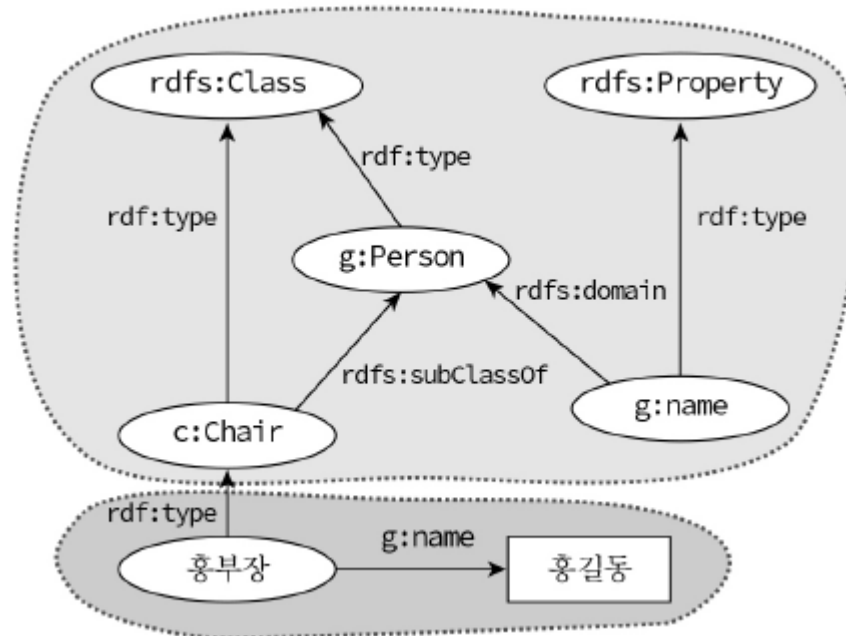
```
<rdfs:Class rdf:ID="http://schema.org/given#Person">
  <rdf:type> <rdfs:Class> </rdf:type>
</rdfs:Class>
<rdfs:Class rdf:ID="http://schema.org/com#Chair">
  <rdfs:subClassOf rdf:resource="http://schema.org/given#Person">
</rdfs:Class>
<rdfs:Property rdf:ID="http://schema.org/given#name">
  <rdfs:domain rdf:resource="http://schema.org/given#Person">
</rdfs:Property>
```



온톨로지

❖ 온톨로지의 지식표현 – cont.

▪ RDFS(RDF 스키마)



<rdf:RDF

xmlns:g="http://schema.org/given" xmlns:c="http://schema.org/com">

<c:Chair rdf:ID="홍부장">

<g:name> "홍길동" </g:name>

</c:Chair>

</rdf:RDF>

온톨로지

❖ 온톨로지의 지식표현

▪ SPARQL

- RDF 형태로 저장된 데이터에 대한 질의어
- SQL 유사한 문법

PREFIX contact: <http://www.w3.org/2000/10/swap/pim/contact#>

PREFIX rdf: < http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?name ?email

WHERE { ?person **rdf:type** contact:Person.

?person **contact:name** ?name.

?person **contact:mbox** ?email. }

온톨로지

❖ 온톨로지의 지식표현

- RIF (Rule Interchange Format)
 - 규칙을 정의하고 교환하기 위한 규약

```
Prefix(ex <http://example.com/2008/prd1#>)
(* ex:rule_1 *)
Forall ?customer ?purchasesYTD (
  If And ( ?customer#ex:Customer
    ?customer[ex:purchasesYTD->?purchasesYTD]
    External(pred:numeric-greater-than(?purchasesYTD 5000)) )
  Then Do ( Modify(?customer[ex:status->"Gold"]) ) )
```

당해년도 누적 구매금액이 \$5,000이상이면 “Gold” 등급으로 조정

온톨로지

❖ OWL (Web Ontology Language)

- 웹 상의 **자원**과 이들의 **속성**에 대한 **지식**을 **표현**하기 위한 온톨로지 언어
- 예
 - Vintage라는 자원이 하나의 hasVintageYear라는 속성을 하나만 갖는다는 것을 표현

```
<owl:Class rdf:ID="Vintage">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasVintageYear"/>
      <owl:cardinality
        rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

온톨로지

❖ 온톨로지의 활용

▪ 시맨틱 웹(semantic web)

- 웹의 데이터를 **소프트웨어 에이전트**가 **이해**하여 지능적으로 활용할 수 있도록 하는 것
- 웹의 처음 설계한 Tim Bernes-Lee가 주창한 아이디어
- **의미있는 태그**(tag)를 정의하여 문서를 기술하기 위해 XML 사용
- 태그 및 데이터의 **의미 해석**을 위해 **RDF** 사용
- **온톨로지** 구축을 통해 태그 및 메타 데이터의 의미 해석

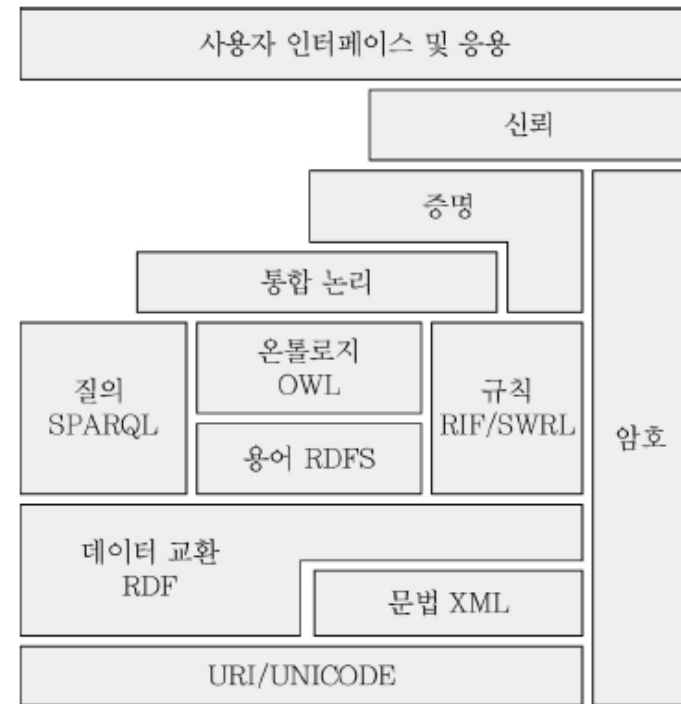


그림 3.15 W3C 시맨틱 웹 기술 스택

온톨로지

❖ 의미망과 온톨로지

- 그래프 구조를 이용 지식 모델링

- 의미망

- 대상, 관계 등의 표현에 사용되는 용어가 임의적
- 통일된 표현 관련 규정 부재

- 온톨로지

- 다른 시스템과 공유 및 상호운영성(interoperability)을 위해 명확한 지침에 따라 표현
- 구축된 정보 및 지식의 재사용에 관심

온톨로지와 RDF

❖ 온톨로지 구축 도구

▪ Protégé 등

- 지식 획득 및 표현을 위한 GUI 기반의 오픈 소스 온톨로지 편집기

The image displays the Protégé ontology editor interface. On the left, the 'Classes' browser shows a hierarchical tree of ontology classes. The 'annulate lamellae' class is selected and highlighted. On the right, the 'Class description for annulate lamellae' window provides details for this class.

Classes

- owl:Thing
 - biological_process
 - cellular_component
 - axon part
 - axoneme part
 - bacterial-type flagellum part
 - cell
 - cell cortex part
 - cell division site part
 - cell junction
 - cell part
 - annulate lamellae**
 - apical complex
 - apical part of cell
 - ascus epiplasm
 - aspartate carbamoyltransferase complex
 - basal labyrinth
 - basal part of cell
 - basal ring of apical complex
 - CD20-Lck-Fyn complex
 - CD20-Lck-Lyn-Fyn complex
 - cell body
 - cell division site
 - cell division site part
 - cell envelope Sec protein transport complex
 - cell leading edge
 - cell periphery
 - cell pole
 - cell projection
 - cell projection part
 - cell septum
 - cell septum part

Class description for annulate lamellae

Display name
annulate lamellae

IRI
http://purl.obolibrary.org/obo/GO_0005642

Annotations

- rdfs:label: annulate lamellae
- id: GO:0005642
- definition: Stacks of endoplasmic reticulum (ER) membranes containing a high density of nuclear pores, thought to form from excess nuclear membrane components, that have been described in a number of different cells. Annulate lamellar membranes are continuous with and embedded within the ER.
- has_obo_namespace: cellular_component

Properties

- part_of: nuclear outer membrane-endoplasmic reticulum membrane network

4. 지식 그래프

❖ 지식 그래프(knowledge graph)

- 그래프 구조를 이용하여 **개체**(entity), 개체 간의 **관계**(relationship), 개체의 속성 등을 표현한 것
 - 노드(node) : 개체
 - 간선(edge) : 관계

▪ 의미망의 형태



세종대왕

검색결과 약 3,010,000개 (0.45초)

세종대왕 관련 이미지

한글 집현전 훈민정음 고화질 업적 문명

조선 세종 국왕

세종은 조선의 제 4대 국왕이며 언어학자이다. 그의 업적에 대한 존경의 의미를 담은 명칭인 세종대왕으로 자주 일컬어진다. 성씨는 이씨, 이름은 도, 본관은 전주, 자는 원정, 아명은 막동이었다. 묘호는 세종이며, 시호는 영문예무인성명효대왕이고, 명나라에서 받은 시호는 장헌이었다. 위키백과

출생: 1397년 5월 15일, 서울특별시
사망 날짜/장소: 1450년 3월 30일, 한성부
본명: 이도
자녀: 조선 세조, 조선 문종, 안평대군, 광평대군, 정소공주, 임영대군, 더보기
형제자매: 양녕대군, 효령대군, 성녕대군, Prince Huryeong, 경녕군, 더보기
배우자: 소헌왕후 (1408년~1446년), 소용 홍씨, 혜빈 양씨, 상침 송씨, 신빈 김씨, 사기 자씨

함께 찾은 검색어 15개 이상 항목 더보기

조선 태종 조선 세조 조선 태조 조선 문종

ko.wikipedia.org › wiki › 조선_세종

조선 세종 - 위키백과, 우리 모두의 백과사전

그러나 세종대왕은 부왕이 숨겨둔 책들을 찾아내어 독서를 계속하였다. 대군 진봉, 그리고 형들과의 경쟁[편집]. 1412년 16살에 둘째 ...

사망지: 조선 한성부 영응대군 사저 동별궁
재위: 1418년 9월 9일 - 1450년 3월 30일
문종 · 태종 · 소헌왕후

휘: 이도(李瑈)
시호: 장헌영문예무인성명효대왕(莊憲英文...

피드백

지식 그래프

❖ 지식 그래프의 구성

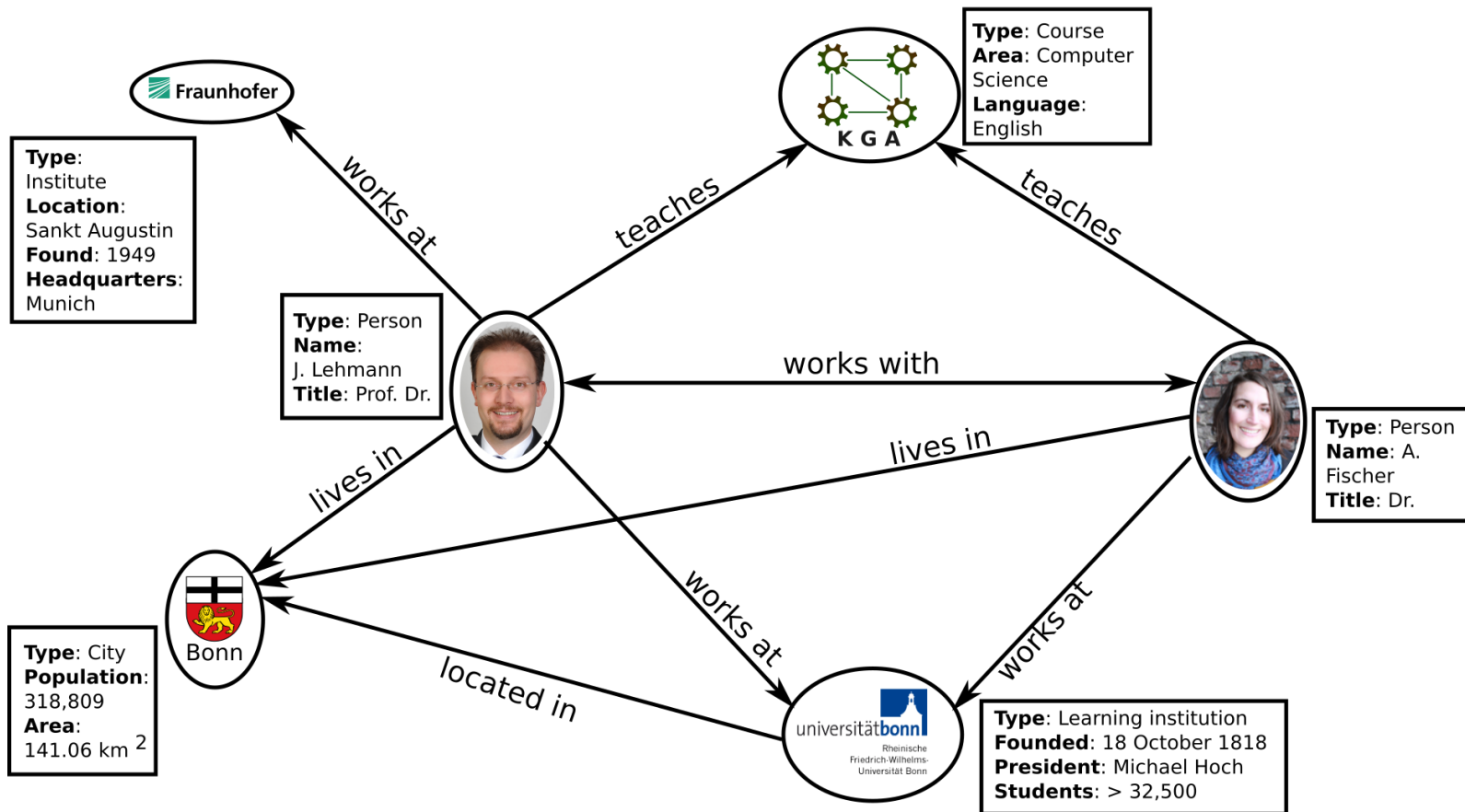
- 다양한 정보 소스에서 추출된 구조화된 정보와 반구조화된 정보를 통합하여 연결된 데이터의 집합
 - **개체(entity)**
 - 지식 그래프 내의 기본적인 개체나 개체
 - 사람, 장소, 사물, 아이디어 등
 - **관계(relation)**
 - 개체 간의 연결 정보
 - 예. 스티브 잡스는 애플의 **공동 창업자**였다 (개체: 스티브 잡스, 애플, 관계: 공동창업자)
 - **속성(attribute)**
 - 개체에 대한 구체적인 정보나 특성
 - 예. 애플은 1976년에 **설립**되었다 (개체: 애플, 속성: 설립연도)

지식 그래프

❖ 지식 그래프의 활용

- 다양한 데이터 소스의 통합
 - 여러 데이터 소스에서 정보를 추출하고 통합, 연결하여 보다 광범위하고 다양한 지식 제공
- 시맨틱 검색
 - 사용자의 질의를 단순한 키워드 검색을 넘어서 해당 질의의 의미와 관련된 정보를 제공
 - 구글의 지식 그래프
 - 검색시 특정 주제나 개체에 대한 요정정보 박스 제공
- 정보의 연결성과 의미를 중심으로 한 지식 탐색
- 추천시스템
- 질의응답 시스템

지식 그래프



프로그래밍 실습: 지식 그래프

❖ spaCy 패키지

- 자연어처리를 위한 오픈 소스 기반 라이브러리로 텍스트 전처리에 유용
- 토큰화, 품사 태킹, 파싱, 원형복원, 문장경계검출, 유사도 계산, 분류, 매칭(matching), 개체명 인식(NER)

```
1 !pip install spacy
```

```
1 import spacy
2 nlp = spacy.load('en_core_web_sm') // Small English pipeline model for tok2vec, tagger, parser,
3                                     // sender, NER, attribute_ruler, lemmatizer
4 doc = nlp("I have seen the handsome boy with an apple.")
5
6 for tok in doc:
7     print(tok.text, tok.lemma_, tok.pos_, tok.tag_, tok.dep_, tok.shape_,
8           tok.is_alpha, tok.is_stop)
```

Text	Lemma	POS	Tag	Dep	Shape	isAlpha	isStop
I	-PRON-	PRON	PRP	nsubj	X	True	True
have	have	AUX	VBP	aux	xxxx	True	True
seen	see	VERB	VBN	ROOT	xxxx	True	False
the	the	DET	DT	det	xxx	True	True
handsome	handsome	ADJ	JJ	amod	xxxx	True	False
boy	boy	NOUN	NN	dobj	xxx	True	False
with	with	ADP	IN	prep	xxxx	True	True
an	an	DET	DT	det	x	True	True
apple	apple	NOUN	NN	pobj	xxxx	True	False
.	.	PUNCT	.	punct	.	False	False

- Text: 원래 단어
- Lemma: 기본형
- POS: 품사
- Tag: 구체적인 품사
- Dep: 구문 의존 기호(토큰 간의 관계)
- Shape: 단어 형태(대소문자, 구두점, 숫자)
- isAlpha: 알파벳 여부
- isStop: 중지 단어(stop word) 여부

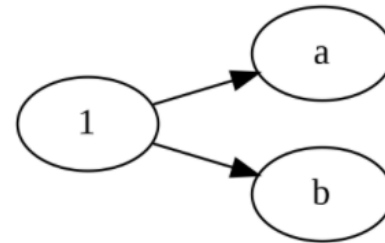
프로그래밍 실습: 지식 그래프

❖ networkx 패키지

- 그래프 등의 네트워크를 생성, 관리, 출력하는 패키지

```
1 import networkx as nx
2 g1 = nx.DiGraph() # 방향 그래프(directed graph)
3 g1.add_node('a') # 노드 추가
4 g1.add_node(1)
5 g1.add_node('b')
6 print('Node: ', g1.nodes())
7 g1.add_edge(1, 'a') # 에지 추가
8 g1.add_edge(1, 'b')
9 print('Edges: ', g1.edges())
10
11 from IPython.core.display import Image
12 from networkx.drawing.nx_pydot import to_pydot
13 d1 = to_pydot(g1)
14 d1.set_dpi(400)
15 d1.set_rankdir('LR')
16 d1.set_margin(1)
17 Image(d1.create_png(), width=300)
```

Node: ['a', 1, 'b']
Edges: [(1, 'a'), (1, 'b')]



프로그래밍 실습: 지식 그래프

❖ 지식 그래프 구성

- 영어 문장들에서 **개체**와 개체 간의 **관계** 추출
 - 문장에서 개체 추출 방법
 - 개체 간의 관계 추출 방법
 - **spaCy** 이용
- 개체들과 관계를 이용하여 **지식 그래프** 구성
 - 개체는 노드, 관계는 간선으로 표현
 - **networkx** 이용

```

1 import re
2 import pandas as pd
3 import bs4
4 import requests
5 import spacy
6 from spacy import displacy
7 nlp = spacy.load('en_core_web_sm')
8
9 from spacy.matcher import Matcher
10 from spacy.tokens import Span
11
12 import networkx as nx
13
14 import matplotlib.pyplot as plt
15 from tqdm import tqdm
16
17 pd.set_option('display.max_colwidth', 200)
18 %matplotlib inline

```

```

1 # wikipedia 문서 파일: https://github.com/phgunawan/Latihan-ML/blob/master/wiki\_sentences\_v2.csv
2 candidate_sentences = pd.read_csv("wiki_sentences_v2.csv")
3 candidate_sentences.shape

```

1	sentence
2	confused and frustrated, connie decides to leave on her own.
3	later, a woman's scream is heard in the distance.
4	christian is then paralyzed by an elder.
5	the temple is set on fire.
6	outside, the cult wails with him.
7	it's a parable of a woman's religious awakening—
8	c. mackenzie, and craig vincent joined the cast.

```

1 def get_entities(sent):
2     ent1 = ""
3     ent2 = ""
4     prv_tok_dep = ""    # 문장에서 직전 토큰의 의존 파싱 태그
5     prv_tok_text = ""   # 문장에서 직전 토큰
6     prefix = ""
7     modifier = ""
8     for tok in nlp(sent):
9         # 토큰이 구두점(punctuation mark)이면 다음 토큰으로 이동
10        if tok.dep_ != "punct":
11            if tok.dep_ == "compound": # 토큰이 복합어인 경우
12                prefix = tok.text
13                if prv_tok_dep == "compound": # 직전 토큰이 복합어이면 현재 토큰과 결합
14                    prefix = prv_tok_text + " " + tok.text
15            if tok.dep_.endswith("mod") == True: # 토큰이 수식어(modifier)인 경우
16                modifier = tok.text
17                if prv_tok_dep == "compound": # 직전 토큰이 수식어이면 현재 토큰을 결합
18                    modifier = prv_tok_text + " " + tok.text
19
20            if tok.dep_.find("subj") == True: # 주어(subject)인 경우,
21                ent1 = modifier + " " + prefix + " " + tok.text # 수식어와 현재 토큰 결합 => 개체명 생성
22                prefix = ""
23                modifier = ""
24                prv_tok_dep = ""
25                prv_tok_text = ""
26            if tok.dep_.find("obj") == True: # 목적어인 경우
27                ent2 = modifier + " " + prefix + " " + tok.text # 수식어와 현재 토큰 결합 => 개체명 생성
28
29            prv_tok_dep = tok.dep_
30            prv_tok_text = tok.text
31    return [ent1.strip(), ent2.strip()]    # 식별된 개체명 반환

```

```
1 get_entities("the film had 200 patents")
```

```
['film', '200 patents']
```

```
1 entity_pairs = []
2
3 for i in tqdm(candidate_sentences["sentence"]):
4     entity_pairs.append(get_entities(i))
```

```
1 entity_pairs[10:20]
```

```
[['we', 'tests'],
 ['m global', 'international sales rights'],
 ['robbie robertson', 'soundtrack'],
 ['it', 'original music tracks'],
 ['it', 'reviewed franchise'],
 ['she', 'accidentally mystique'],
 ['', 'military arrest'],
 ['train', 'vuk'],
 ['telepath', 'gallio'],
 ['singer', 'men']]
```

```

1 def get_relation(sent):
2     doc = nlp(sent)
3     matcher = Matcher(nlp.vocab)
4
5     # 패턴 정의
6     pattern = [{'DEP': 'ROOT'},
7                 {'DEP': 'prep', 'OP': "?"},
8                 {'DEP': 'agent', 'OP': "?"},
9                 {'POS': 'ADJ', 'OP': "?"}]
10
11     matcher.add("matching_1", None, pattern)
12
13     matches = matcher(doc)
14     print('matches :', matches)
15     k = len(matches) - 1
16
17     span = doc[matches[k][1]:matches[k][2]]
18
19     return(span.text)

```

```

1 get_relation("John completed the task")

```

matches :
'completed'

```

1 relations = [get_relation(i) for i in tqdm(candidate_sentences['sentence'])]

```



```

1 # 주어(subject) 추출
2 source = [i[0] for i in entity_pairs]
3
4 # 목적어(object) 추출
5 target = [i[1] for i in entity_pairs]
6
7 kg_df = pd.DataFrame({'source':source, 'target':target, 'edge':relations})

```

```

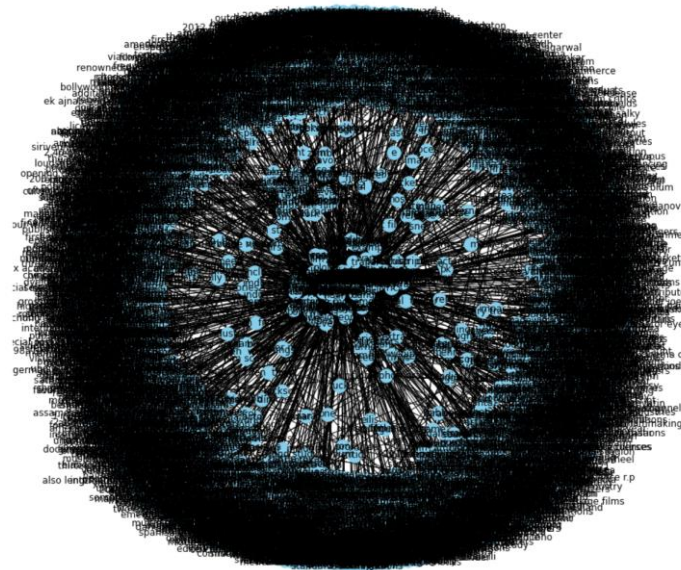
1 # 방향 그래프 생성
2 G = nx.from_pandas_edgelist(kg_df, "source", "target",
3                             edge_attr=True, create_using=nx.MultiDiGraph())

```

```

1 # 그래프 그리기
2 plt.figure(figsize=(12,12))
3 pos = nx.spring_layout(G)
4 nx.draw(G, with_labels=True, node_color='skyblue', edge_cmap=plt.cm.Blues, pos = pos)
5 plt.show()

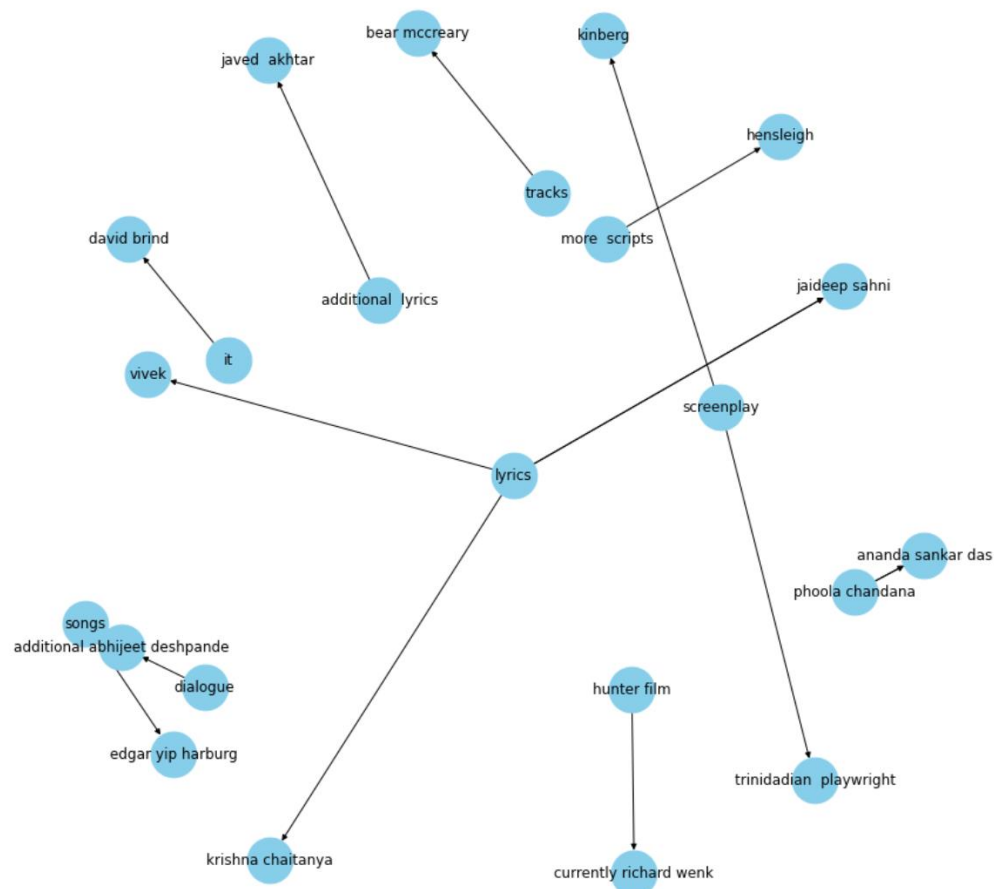
```



```

1 G=nx.from_pandas_edgelist(kg_df[kg_df['edge']=="written by"], "source", "target",
2                           edge_attr=True, create_using=nx.MultiDiGraph())
3
4 plt.figure(figsize=(12,12))
5 pos = nx.spring_layout(G, k = 0.5)
6 nx.draw(G, with_labels=True, node_color='skyblue', node_size=1500, edge_cmap=plt.cm.Blues, pos = pos)
7 nx.draw_networkx_edge_labels(G, pos, edge_labels="written by")
8 plt.show()

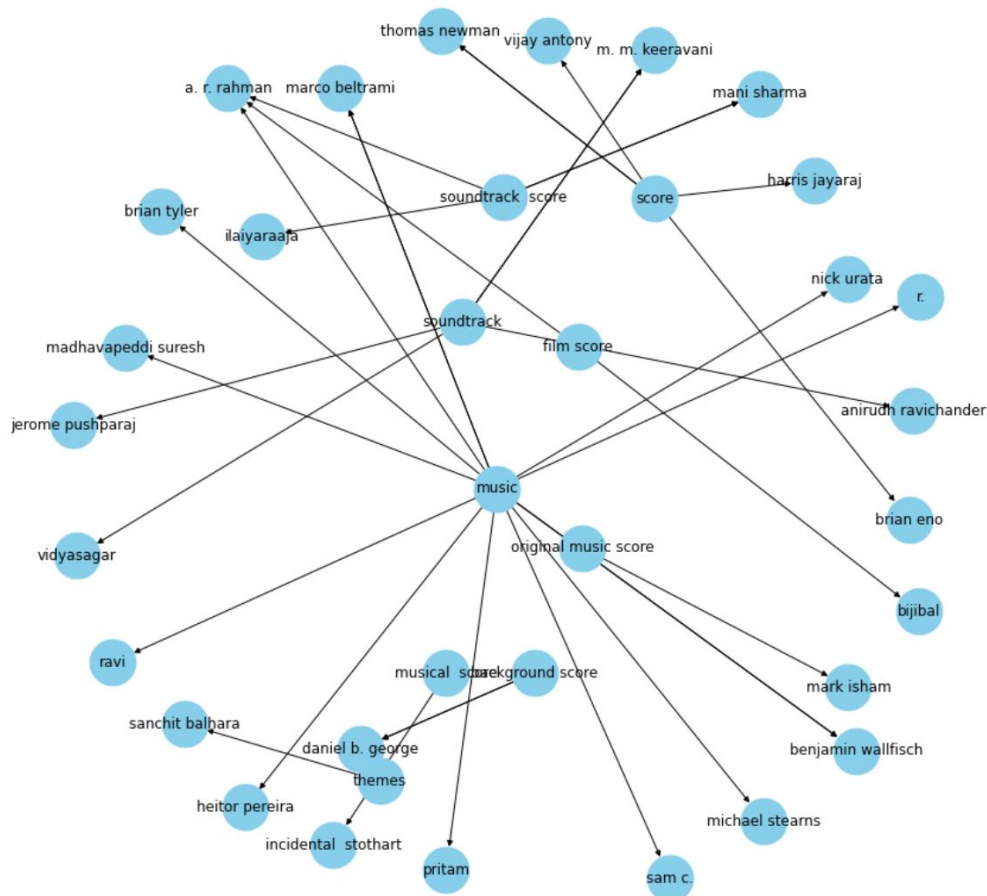
```



```

1 G=nx.from_pandas_edgelist(kg_df[kg_df['edge']=="composed by"], "source", "target",
2                             edge_attr=True, create_using=nx.MultiDiGraph())
3
4 plt.figure(figsize=(12,12))
5 pos = nx.spring_layout(G, k = 0.5) # k regulates the distance between nodes
6 nx.draw(G, with_labels=True, node_color='skyblue', node_size=1500, edge_cmap=plt.cm.Blues, pos = pos)
7 plt.show()

```



Quiz

❖ 다음 의미망에 대한 설명으로 옳지 않은 것을 선택하시오.

- ① 의미망은 이항 관계의 지식을 표현할 수 있다.
- ② is-a 관계는 추이적 관계이다.
- ③ 다항 관계를 의미망으로 표현하기 위해서는 사물화를 통해 이항 관계로 변환한다.
- ④ 의미망은 프레임과 같은 상속의 개념을 사용하지 않는다.

❖ 다음 의미망과 프레임에 대한 설명으로 옳지 않은 것을 선택하시오.

- ① 의미망에 대한 추론을 할 때는 질문에 해당하는 관계를 의미망에서 찾고, 이를 통해 알고자 하는 대상에 대한 정보를 찾는다.
- ② 의미망은 프레임으로 변환할 수 있다.
- ③ 의미망을 통해 개념의 계층적 관계를 표현할 수 있다.
- ④ 의미망은 기본적으로 동적인 지식을 표현하기 때문에 쉽게 구조를 바꿀 수 있다.

Quiz

❖ 의미망(semantic network)의 특징 중 틀린 것은 무엇인가?

- ① 노드와 연결선으로 정보를 표현한다.
- ② 개념 간의 관계를 시각화하는데 사용된다.
- ③ 주로 통계적인 정보 처리에 사용된다.
- ④ 계층적 구조나 비계층적 구조 모두를 표현할 수 있다.

❖ 스크립트에 대한 설명으로 옳바르지 않은 것은?

- ① 사람들이 특정 상황에서 예상하는 행동의 순서나 패턴을 나타낸다.
- ② 스크립트는 사건의 기본적인 구조를 이해하는 데 도움을 준다.
- ③ 스크립트는 항상 논리적이고 합리적인 행동만을 포함한다.
- ④ 일상적인 상황에서 어떤 행동이나 반응이 일어날 것인지 예측하는 데 사용된다.

Quiz

❖ 다음 스크립트에 대한 설명으로 옳지 않은 것은?

- ① 진입조건에는 기술된 사건들이 발행하기 전에 만족해야 할 것들이다.
- ② 장면은 실제 일어나는 일련의 사건을 나타낸다.
- ③ 스크립트에는 일련의 사건을 표현하기 위해 선언적인 지식을 포함할 수 있다.
- ④ 역할자는 스크립트의 실행에 관련된 사람들을 포함한다.

❖ 다음 온톨로지에 대한 설명으로 옳지 않은 것은?

- ① 온톨로지는 특정 영역의 개념과 관계를 표현하기 위해 사용될 수 있다.
- ② 특정 온톨로지를 기반으로 표현된 지식은 공유하기 편리하다.
- ③ RDF 형식으로 온톨로지를 표현할 수 있다.
- ④ SPARQL은 SQL과 유사한 문법으로 메타데이터를 기술하는데 사용할 수 있다.

Quiz

❖ 온톨로지의 주요 목적은 무엇인가요?

- ① 데이터 암호화
- ② 웹 페이지 디자인
- ③ 개념 및 개념 간의 관계 표현
- ④ 알고리즘 최적화

❖ 온톨로지는 다음 중 어떤 분야와 밀접한 관련이 있나요?

- ① 음악 작곡
- ② 시맨틱 웹
- ③ 그래픽 디자인
- ④ 애니메이션 제작

❖ 온톨로지의 핵심 구성 요소가 아닌 것은 무엇인가요?

- ① 클래스
- ② 속성
- ③ 인스턴스
- ④ 알고리즘

Quiz

❖ **RDF(Resource Description Framework)에 대한 설명으로 옳지 않은 것은 무엇인가요?**

- ① RDF는 웹 자원에 대한 설명을 위한 프레임워크이다.
- ② RDF는 주제, 서술어, 목적어의 3요소로 구성된다.
- ③ RDF는 데이터의 구조와 내용을 동시에 표현하는 XML 기반의 언어이다.
- ④ RDF는 데이터 간의 관계를 묘사하는데 사용되는 XML 문서 형식이다.

❖ **온톨로지의 정의에 가장 가까운 것은?**

- ① 웹 페이지의 디자인과 레이아웃을 위한 도구
- ② 데이터베이스의 테이블과 열을 정의하는 구조
- ③ 특정 도메인의 개체와 그들 사이의 관계를 정의하고 분류하는 공식적인 사양
- ④ 프로그래밍 언어의 문법과 구문을 정의하는 규칙

❖ **RDF 스키마(RDFS)의 주요 기능 중 하나는 무엇인가요?**

- ① 웹 페이지 디자인
- ② 데이터 압축
- ③ 리소스 간의 관계와 클래스 구조 기술
- ④ 암호화

Quiz

❖ **RDF에서 각 리소스를 고유하게 식별하기 위해 사용하는 것은 무엇인가요?**

- ① URL
- ② UDI
- ③ URN
- ④ URI

❖ **온톨로지를 개발하고 관리하기 위한 인기 있는 언어로 알려진 것은?**

- ① SQL
- ② OWL (Ontology Web Language)
- ③ HTML
- ④ CSS

❖ **지식 그래프(Knowledge Graph)의 주요 특성 중 어느 것이 옳지 않은가?**

- ① 지식 그래프는 연결된 데이터의 모음으로 개체, 속성, 관계 등을 포함한다.
- ② 지식 그래프는 시맨틱 웹의 핵심 구성 요소 중 하나이다.
- ③ 지식 그래프는 주로 정적 데이터만을 다룬다.
- ④ 지식 그래프는 복잡한 질의 응답, 추천 시스템, 인공지능 등의 애플리케이션에 활용될 수 있다.

Quiz

❖ 구글의 지식 그래프는 어떤 목적으로 개발되었나?

- ① 광고 매출을 늘리기 위해
- ② 사용자 검색 쿼리에 대해 더 풍부하고 정확한 정보를 제공하기 위해
- ③ Google Drive의 저장 용량을 확장하기 위해
- ④ Gmail의 사용자 인터페이스를 개선하기 위해

❖ 지식 그래프의 장점 중 어느 것이 옳지 않은가?

- ① 다양한 데이터 소스 간의 연결을 통한 통합된 지식 제공
- ② 복잡한 질의에 대한 응답을 지원
- ③ 사용자의 검색 경험 향상
- ④ 데이터의 중복성을 증가시키는 구조