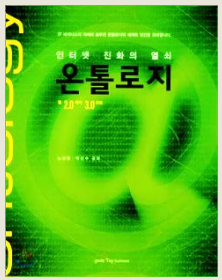



인 터 넷 진 화 의 열 쇠

# 온 톨 로 지

웹 2.0에서 3.0으로

노상규·박진수 공저

god's toy business, 2007

SNU IDB laboratory

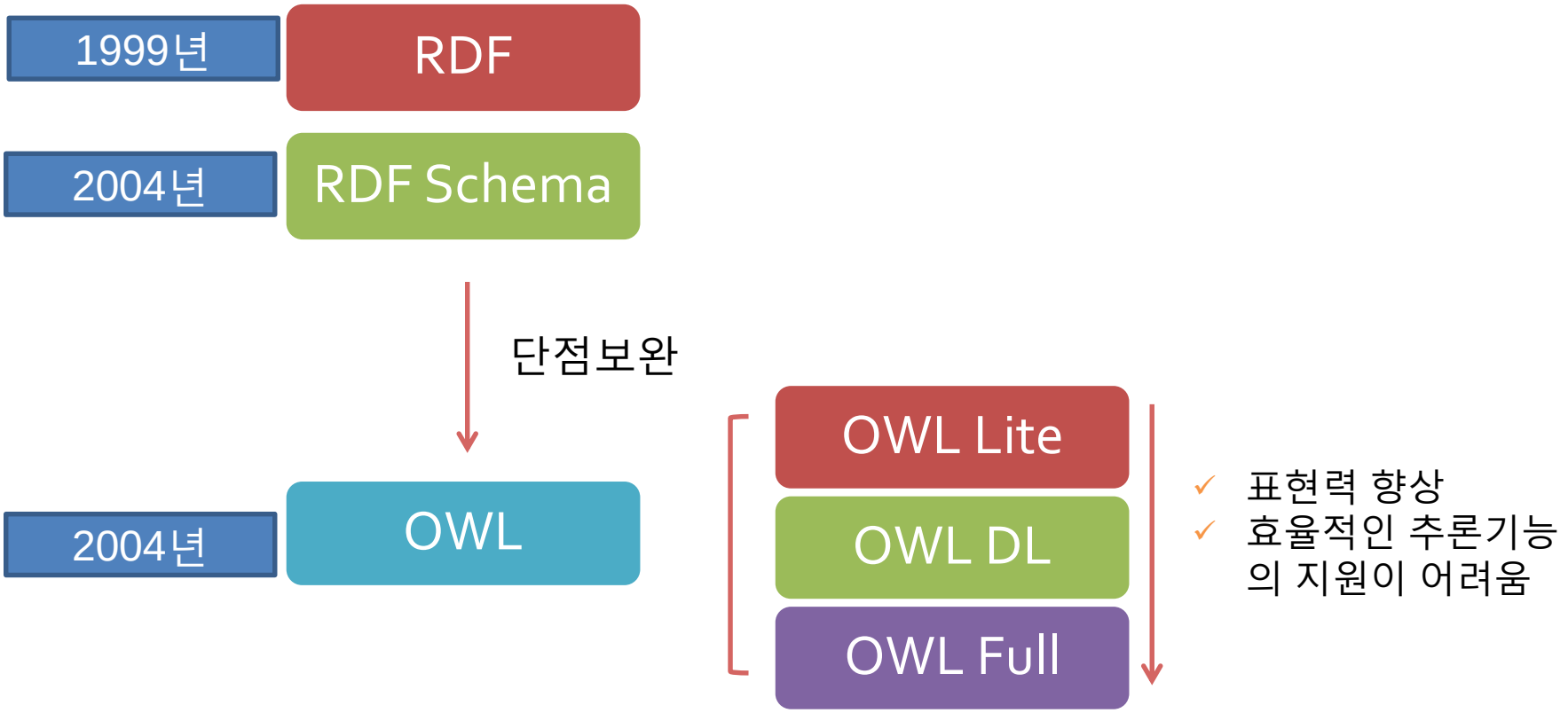
# Ontology Contents

## MODULE 1 온톨로지의 개념 및 응용



## MODULE 2 온톨로지 언어와 구축도구

# Chapter 7 OWL(Web Ontology Language)



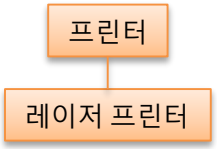
# Chapter 7 OWL(Web Ontology Language)

- 1. OWL의 기본 요소 : 클래스와 속성
  - 1.1 클래스, 개체 및 클래스 계층구조
  - 1.2 속성과 계층구조
- 2. OWL의 새로운 기능
  - 2.1 클래스와 속성의 표현
  - 2.2 특별한 속성을 통한 추론 기능의 활성화
  - 2.3 온톨로지 병합과 재사용
- 3. 세 종류의 OWL
  - 3.1 OWL Lite
  - 3.2 OWL DL(Description Logic)
  - 3.3 OWL Full
- 4. OWL 예제

## 1. OWL의 기본 요소 : 클래스와 속성

# 1.1 클래스, 개체 및 클래스 계층구조

- ‘rdf:’ 혹은 ‘rdfs:’를 앞에 붙이지 않은것은 OWL의 namespace를 사용
- owl: Class
  - OWL Full에서는 rdfs:Class와 동일
  - OWL Lite와 OWL DL에서는 모든 refs:Class가 owl:Class가 아님
- owl: Thing ➔ 모든 instance의 최상위 class
- owl: Nothing ➔ 포함하는 instance가 하나도 없는 class
- rdfs:subClassOf
  - 예) 레이저 프린터는 프린터의 하위 클래스이다

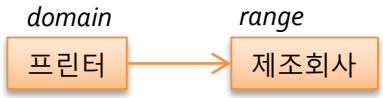



```
<owl:Class rdf:ID = “레이저 프린터”>
    <rdfs:subClassOf rdf:resource=“#프린터”/>
</owl:Class>
```

- Class의 extension(외연)과 intension(내포)
  - ‘English101 수강생‘과 ‘Calculus105 수강생’의 구성원 같을때 ➔
  - 두 class의 extension은 같다, 드러나 두 class의 intension은 다르고 ➔
  - 두 class는 동일하지 않다

# 1. OWL의 기본 요소 : 클래스와 속성
# 1.2 속성과 계층구조 (1/2)

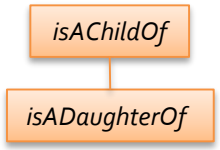
- Owl:ObjectProperty ➔ 클래스의 인스턴스를 다른 클래스에 속한 인스턴스와 연결하는 속성
  - *rdfs:domain*, *rdfs:range*로 속성의 정의역(domain)과 치역(range)를 지정
  - 예제 : '프린터' – '제조하다' – '제조회사'




```
<owl:ObjectProperty rdf:ID = "제조하다">
    <rdfs:domain rdf:resource = "#프린터"/>
    <rdfs:range rdf:resource = "#제조회사"/>
</owl:ObjectProperty>
```

- *Owl:DatatypeProperty* ➔ 클래스의 인스턴스를 특정한 데이터타입과 연결하는 속성
- rdfs:subProperty ➔ 속성의 계층을 형성할 때 사용
  - 예) '~의 딸이다(*isADaughterOf*)' 는 '~의 자식이다(*isAChildOf*)'의 하위 속성이다

```
<owl:ObjectProperty rdf:ID = "isADaughterOf">
    <rdfs:subProperty rdf:resource="#isAChildOf"/>
</owl:ObjectProperty>
```



# Chapter 7 OWL(Web Ontology Language)

- 1. OWL의 기본 요소 : 클래스와 속성
  - 1.1 클래스, 개체 및 클래스 계층구조
  - 1.2 속성과 계층구조
- 2. OWL의 새로운 기능
  - 2.1 클래스와 속성의 표현
  - 2.2 특별한 속성을 통한 추론 기능의 활성화
  - 2.3 온톨로지 병합과 재사용
- 3. 세 종류의 OWL
  - 3.1 OWL Lite
  - 3.2 OWL DL(Description Logic)
  - 3.3 OWL Full
- 4. OWL 예제

# 2. OWL의 새로운 기능

- OWL은 추가적인 어휘를 통해 RDF(S)의 표현력을 확장
  - 불(Boolean)연산을 통한 복잡한 클래스
  - 열거형 클래스 표현
  - 클래스의 비접합성(disjointness)
  - 속성에 대한 다양한 범위
  - 관계 차수(cardinality)
  - 특별한 속성
  - 동치성(equality)과 비동치성(inequality)
  - 온톨로지 버저닝(versioning)

1) 클래스와 속성의 표현

2) 추론 기능

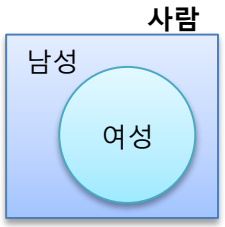
3) 온톨로지 재사용

## 2. OWL의 새로운 기능
# 2.1 클래스와 속성의 표현 (1/8)

- 클래스의 불(Boolean) 연산
  - *owl:intersectionOf*, *owl:unionOf*
    - 클래스 간의 교집합, 합집합으로 새로운 클래스를 정의
  - *owl:complementOf*
    - 한 클래스가 다른 한 클래스의 여집합이라는 것을 나타냄
  - 예) '사람'은 '여성'과 '남성'의 합집합

    ➔ '사람'중에서 '여성'에 속하지 않는 모든 인스턴스는 '남성'

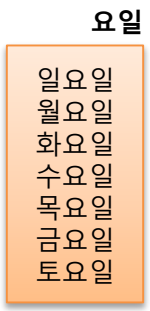
```
<owl:Class rdf:ID = "사람"
    <owl:unionOf>
        <owl:Class rdf:ID = "여성"/>
        <owl:Class rdf:ID = "남성">
            <owl:complementOf rdf:resource="#여성"/>
        </owl:Class>
    </owl:unionOf>
</owl:Class>
```



## 2. OWL의 새로운 기능
# 2.1 클래스와 속성의 표현 (2/8)

- 열거형 클래스(Enumerated Class)
  - *Owl:oneOf*
    - 인스턴스가 나열 된 클래스
  - *Owl:Thing*
    - 클래스에 속한 각각의 인스턴스
  - 예) '요일'이라는 열거형 클래스 안에 '일요일' , '월요일', … , '토요일' 이라는 인스턴스가 있다

```
<owl:Class rdf:ID = "요일">
    <owl:oneOf rdf:parseType = "Collection">
        <owl:Thing rdf:about="#일요일"/>
        <owl:Thing rdf:about="#월요일"/>
        <owl:Thing rdf:about="#화요일"/>
        <owl:Thing rdf:about="#수요일"/>
        <owl:Thing rdf:about="#목요일"/>
        <owl:Thing rdf:about="#금요일"/>
        <owl:Thing rdf:about="#토요일"/>
    </owl:oneOf>
</owl:Class>
```



# 2. OWL의 새로운 기능
# 2.1 클래스와 속성의 표현 (3/8)

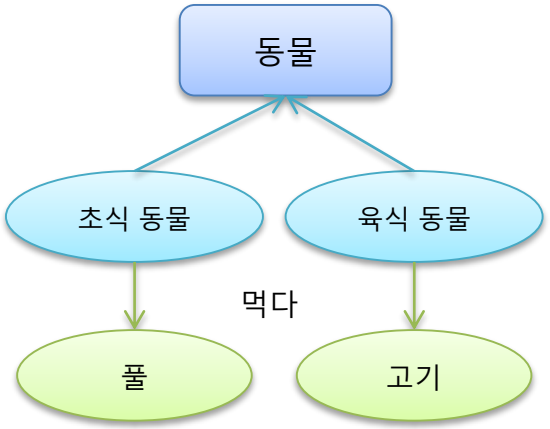
- 클래스의 비접합성
  - *owl:disjointWith*
    - 2개 이상의 클래스가 서로 공통의 인스턴스를 가지지 않음을 표현
  - 예) '레이저 프린터' 와 '잉크젯 프린터' 는 공통의 인스턴스가 존재할 수 없는 클래스이다

```
<owl:Class rdf:ID = "레이저 프린터">
    <owl:disjointWith rdf:resource="#잉크젯 프린터"/>
</owl:Class>
```

## 2. OWL의 새로운 기능

# 2.1 클래스와 속성의 표현 (4/8)

- 속성에 대한 다양한 범위 지정
  - RDF Schema
    - 서로 다른 클래스가 하나의 속성에 대해서 각기 다른 속성값의 범위를 가지게 하는 것이 불가능




**RDF 트리플**

**(초식동물) – (먹다) – (풀)**
**(육식동물) – (먹다) – (고기)**

하나의 속성에 대해
각기 다른 정의역과 공역을
설정하는것은 모순!

## 2. OWL의 새로운 기능
# 2.1 클래스와 속성의 표현 (5/8)

- OWL
  - 온톨로지 전체에서 유효한 정의역과 공역을 명시

```
<owl:ObjectProperty rdf:ID = "먹다">
    <rdfs:domain rdf:resource="#동물"/>
</owl:ObjectProperty>
```

    - 공역을 열어놓고 해당 클래스에서 이를 제한

  - 하나의 정의역과 공역을 설정하면서도 클래스마다 속성값의 범위를 다르게 지정 가능

```
<owl:Class rdf:ID = "초식 동물">
    <rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty rdf:resource = "#먹다"/>
        <owl:allValuesFrom rdf:resource = "#풀"/>
    </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

## 2. OWL의 새로운 기능
# 2.1 클래스와 속성의 표현 (6/8)

- *owl:onProperty*
  - 특정 속성이 제약을 포함하는 속성이라는 의미
  - *owl:allValuesFrom*
    - onProperty로 지정된 속성에 대해 반드시 특정한 클래스로부터만 값을 취한다는 제약 조건
  - *owl:someValuesFrom*
    - 특정한 클래스에서 적어도 하나의 값을 가져온다
    - 예) 자동차는 최소한 하나 이상의 엔진을 부속품으로 포함한다

```
<owl:Class rdf:ID = "자동차">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource = "#hasPart"/>
            <owl:someValuesFrom rdf:resource = "#엔진"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

# 2. OWL의 새로운 기능
# 2.1 클래스와 속성의 표현 (7/8)

- *owl:hasValue*
  - 어떤 속성의 값을 특정한 객체로 제한
  - 예) '한국인'이라는 클래스의 모든 인스턴스는 '국적을 가지다(hasNationality)'라는 속성에 대한 값으로 '한국'을 갖는다

```
<owl:Class rdf:ID ="한국인">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource = "#hasNationality"/>
            <owl:hasValue rdf:resource = "#한국"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

# 2. OWL의 새로운 기능
# 2.1 클래스와 속성의 표현 (8/8)

- 관계차수 표현
  - 어떤 클래스의 인스턴스가 특정 속성을 통해 연결될 수 있는 값이 최소(대)한 몇 개인지를 제한
  - *owl:minCardinality* ➔ 최소한 m개 이상의 값을 지님
  - *owl:maxCardinality* ➔ 최대한 n개 이하의 값을 지님
  - *owl:cardinality*
    - *owl:minCardinality* 와 *owl:maxCardinality* 가 같은 것, 즉 m=n개의 값을 지닌다는 것
  - 예) 모든 학생은 최소한 전공을 한 개 이상 가져야 한다

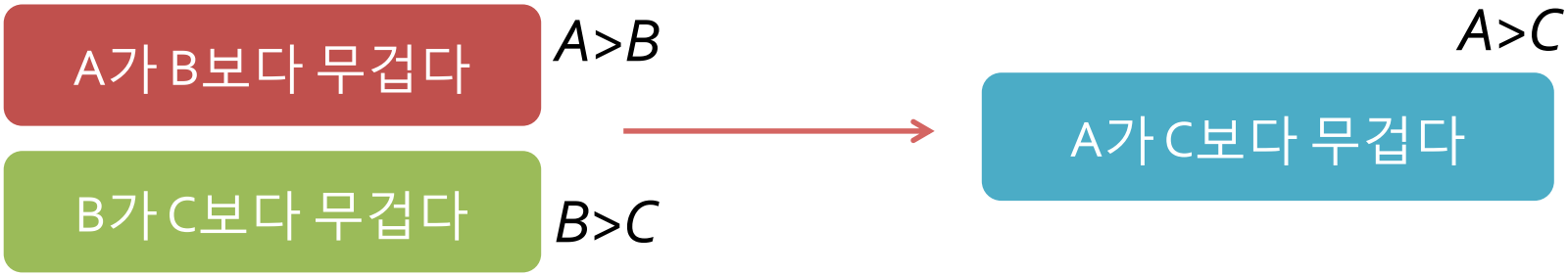
```
<owl:Restriction>
    <owl:onProperty rdf:resource = "#전공하다"/>
    <owl:minCardinality rdf:datatype="nonNegativeInteger"> 1 </owl:minCardinality>
</owl"Restriction>
```

# 2. OWL의 새로운 기능
# 2.2 특별한 속성을 통한 추론 기능의 활성화 (1/6)

- 컴퓨터가 이해할 수 있도록 특별한 의미를 지닌 속성을 나타내는 어휘 사용
  - 이행속성(transitive property)
  - 대칭속성(symmetric property)
  - 함수속성(functional property)
  - 역함수속성(inverse functional property)
  - 역의 관계(inverse of)

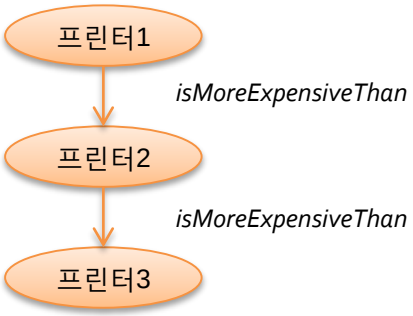
# 2. OWL의 새로운 기능
# 2.2 특별한 속성을 통한 추론 기능의 활성화 (2/6)

- 이행속성(Transitive Property)
  - *owl:TransitiveProperty*



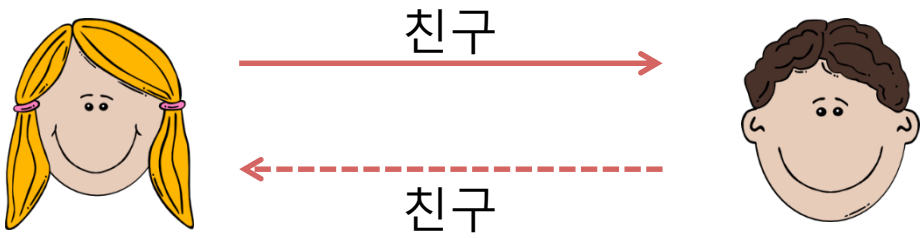

  - 예) '프린터1'이 '프린터2'보다 비싸고, '프린터2'가 '프린터3'보다 비싸다

```
<owl:ObjectProperty rdf:ID = "isMoreExpensiveThan">
    <rdf:type rdf:resource="&owl;TransitiveProperty"/>
</owl:ObjectProperty>

<owl:Class rdf:ID = "프린터1">
    <isMoreExpensiveThan rdf:resource="#프린터2"/>
</owl:Class>

<owl:Class rdf:ID = "프린터2">
    <isMoreExpensiveThan rdf:resource="#프린터3"/>
</owl:Class>
```



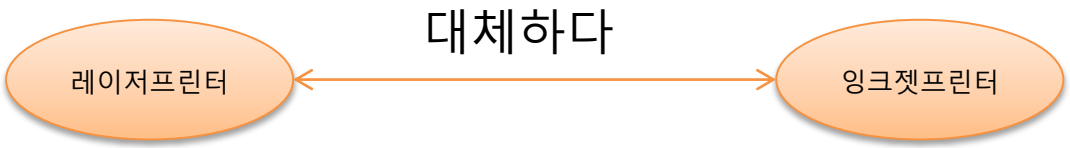
# 2. OWL의 새로운 기능
# 2.2 특별한 속성을 통한 추론 기능의 활성화 (3/6)

- 대칭속성(Symmetric Property)
  - *owl:SymmetricProperty*




  - 예) 레이저 프린터와 잉크젯 프린터는 서로를 대체한다

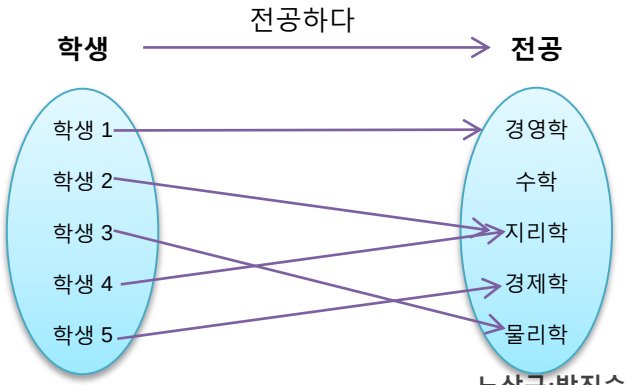
```
<owl:ObjectProperty rdf:ID = "대체하다">
    <rdf:type rdf:resource="&owl;SymmetricProperty"/>
</owl:ObjectProperty>

<owl:Class rdf:ID = "레이저 프린터">
    <대체하다 rdf:resource = "#잉크젯 프린터"/>
</owl:Class>
```



# 2. OWL의 새로운 기능
# 2.2 특별한 속성을 통한 추론 기능의 활성화 (4/6)

- 함수속성
  - *owl:FunctionalProperty*
    - 주어부와 목적부가 항상 함수관계를 이루는 속성
    - 함수관계 : 정의역의 어떠한 자원이 공역에 있는 오직 하나의 값과 연결되는 관계
  - 예) (학생) – (전공하다) – (전공)

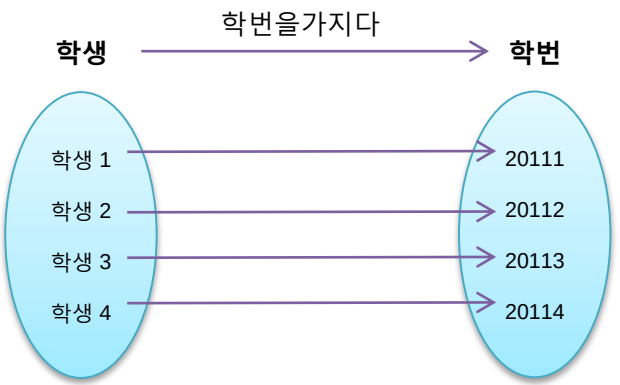
```
<owl:ObjectProperty rdf:ID = "전공하다">
    <rdf:type
rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource = "#학생"/>
    <rdfs:range rdf:resource = "#전공"/>
</owl:ObjectProperty>
```



# 2. OWL의 새로운 기능
# 2.2 특별한 속성을 통한 추론 기능의 활성화 (5/6)

- 역함수속성(Inverse Functional Property)
  - *owl:InverseFunctionalProperty*
    - 속성이 표현하고자 하는 대상은 여러 개의 값을 지닐 수 있으나 그 값은 각각 하나의 대상하고만 연결되어야 한다
    - *owl:FunctionalProperty* 이면서 동시에 *owl:InverseFunctionalProperty* → 일대일 의 관계
  - 예) 학번을 가지다

```
<owl:ObjectProperty rdf:ID = "학번을 가지다">
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource = "#학생"/>
    <rdfs:range rdf:resource = "#학번"/>
</owl:ObjectProperty>
```



# 2. OWL의 새로운 기능
# 2.2 특별한 속성을 통한 추론 기능의 활성화 (6/6)

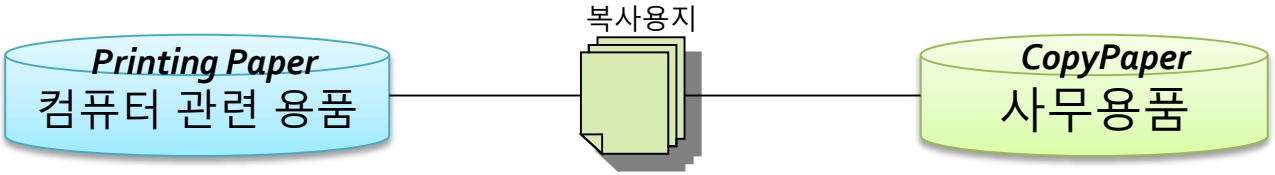
- 역의 관계에 있는 속성(Inverse Of)
  - *owl:InverseOf*
    - 의미상 역의 관계에 있는 속성
  - 예제) '~의 부모이다(*isParentOf*)' ↔ '~의 자식이다(*isChildOf*)'

```
<owl:ObjectProperty rdf:ID = "isParentOf">
    <owl:inverseOf rdf:resource="#isChildOf"/>
</owl:ObjectProperty>
```

# 2. OWL의 새로운 기능
# 2.3 온톨로지 병합과 재사용 (1/3)

- 클래스와 속성 간의 동치성과 비동치성 표현
  - *owl:equivalentClass*
    - 클래스 간의 동치성을 표현

```
<owl:Class rdf:ID = "PrintingPaper">
    <owl:equivalentClass rdf:resource = "&사무용품;CopyPaper">
</owl:Class>
```




    - 여러 개의 클래스와 동치인 경우

```
<owl:Class rdf:about = "#Movie">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <owl:Class rdf:about="#Comedy"/>
                <owl:Class rdf:about="#Drama"/>
                <owl:Class rdf:about="#Thriller"/>
                <owl:Class rdf:about="#Horror"/>
            </owl: unionOf>
        </owl: Class>
    </owl:equivalnetClass>
</owl:Class>
```

# 2.3 온톨로지 병합과 재사용 (2/3)

- *owl:equivalentProperty*
  - 속성간의 동치성을 표현
- *owl:sameAs*
  - 인스턴스간의 동치성을 표현
- *owl:differentFrom*
  - 서로 다른 온톨로지에서 동명인 클래스나 속성, 또는 인스턴스가 각기 다르다는 것을 표현
- *owl:Alldifferent*
  - 하나의 문장으로 여러 개의 클래스나 속성, 인스턴스가 서로 다르다는 사실을 한번에 기술

# 2.3 온톨로지 병합과 재사용 (3/3)

- 온톨로지 버져닝(Versioning)
  - 온톨로지를 병합하거나 재사용할 때, 온톨로지 자체에 대한 정보를 표현
  - *owl:versionInfo*
    - 온톨로지 버전
  - *owl:priorVersion*
    - 특정한 온톨로지가 다른 온톨로지의 이전 버전일 경우
  - *owl:backwardCompatibleWith*, *owl:incompatibleWith*
    - 온톨로지 간의 호환성 여부
  - *owl:DeprecatedClass*, *owl:DeprecatedProperty*
    - 현재 호환성을 위해서 보관되고 있지만 미래에는 없어지게 될 클래스나 속성

# Chapter 7 OWL(Web Ontology Language)

- 1. OWL의 기본 요소 : 클래스와 속성
  - 1.1 클래스, 개체 및 클래스 계층구조
  - 1.2 속성과 계층구조
- 2. OWL의 새로운 기능
  - 2.1 클래스와 속성의 표현
  - 2.2 특별한 속성을 통한 추론 기능의 활성화
  - 2.3 온톨로지 병합과 재사용
- 3. 세 종류의 OWL
  - 3.1 OWL Lite
  - 3.2 OWL DL(Description Logic)
  - 3.3 OWL Full
- 4. OWL 예제

## 3 세종류의 OWL
# 3.1 OWL Lite

- OWL 언어 중 가장 간단한 계층 분류와 제약조건을 표현하기 위한 언어

- 제약
  - 명명되지 않은 클래스 표현에 대한 제약조건이 존재
  - 관계차수의 사용에 제한
    - 관계차수의 값을 오로지 0 또는 1만으로 설정할 수 있음
  - 열거형 클래스, 비접합성 등의 표현이 불가능
  - 클래스, 속성, 개체를 가리키는 URI는 서로 혼용될 수 없음
    - 하나의 자원은 클래스, 속성, 개체 중 오직 한가지로만 표현될 수 있음

# 3 세종류의 OWL
# 3.2 OWL DL(Description Logic)

- 강력한 표현력을 활용하고자 하는 사용자에게 적합
  - 계산상의 완전성(computational completeness) : 모든 결론이 계산될 수 있다는 특성
  - 결정가능성(decidability) : 모든 계산이 유한한 시간 안에 끝난다는 특성

- 제약
  - 임의의 클래스 표현에 대한 제약 조건이 존재
  - 클래스나 속성이 명시적으로 선언되어야 함
  - 속성이 이행속성일 때에는 관계차수를 사용할 수 없음
  - 클래스, 속성, 개체를 가리키는 URI는 서로 혼용될 수 없음
    - 하나의 자원은 클래스, 속성, 개체 중 오직 한가지로만 표현될 수 있음

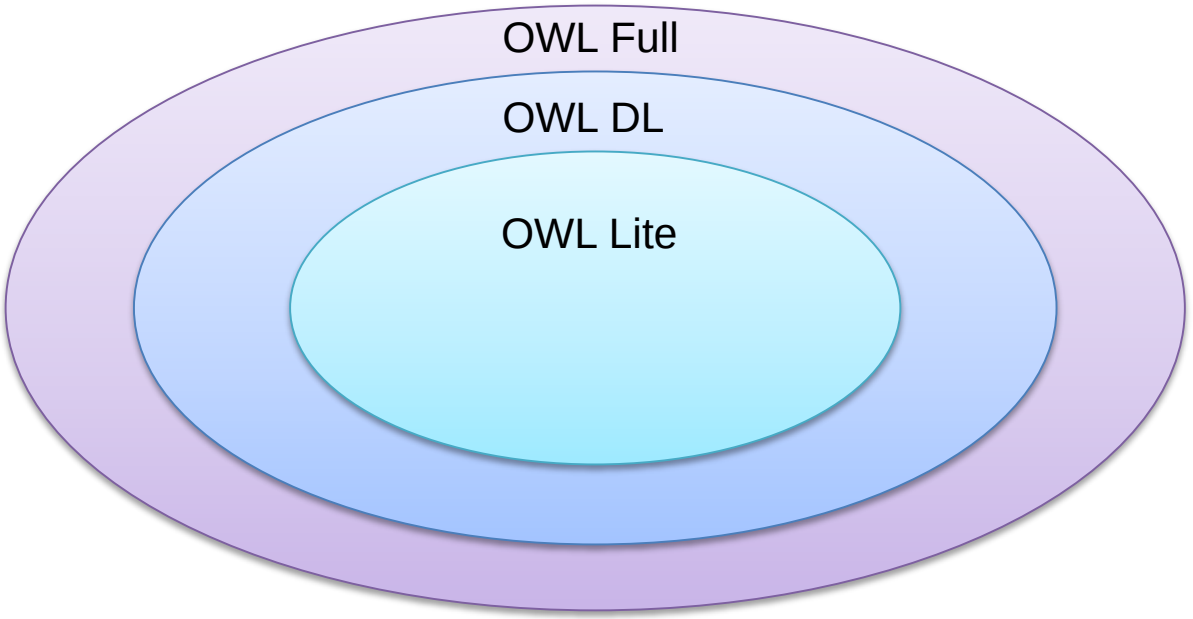
# 3.3 OWL Full

- OWL DL과 동일한 어휘로 구성
- OWL DL 보다 좀 더 자유롭게 어휘를 조합하여 사용
- 모든 기능에 대하여 완전한 추론을 지원하는 추론 SW가 불가능
  - 계산상의 완전성에 대한 보장이 없음
- 최대의 표현력과 RDF의 유연한 문법을 모두 활용

- Lite에서는 배제 됐지만, OWL DL과 OWL Full에서 사용할 수 있는 어휘
  - *oneOf*
  - *disjointWith*
  - *unionOf, complementOf, intersectionOf*
  - *minCardinality, maxCardinality, cardinality*
  - *hasValue*
  - *distinctMembers*

# 3.4 세 종류 OWL의 관계

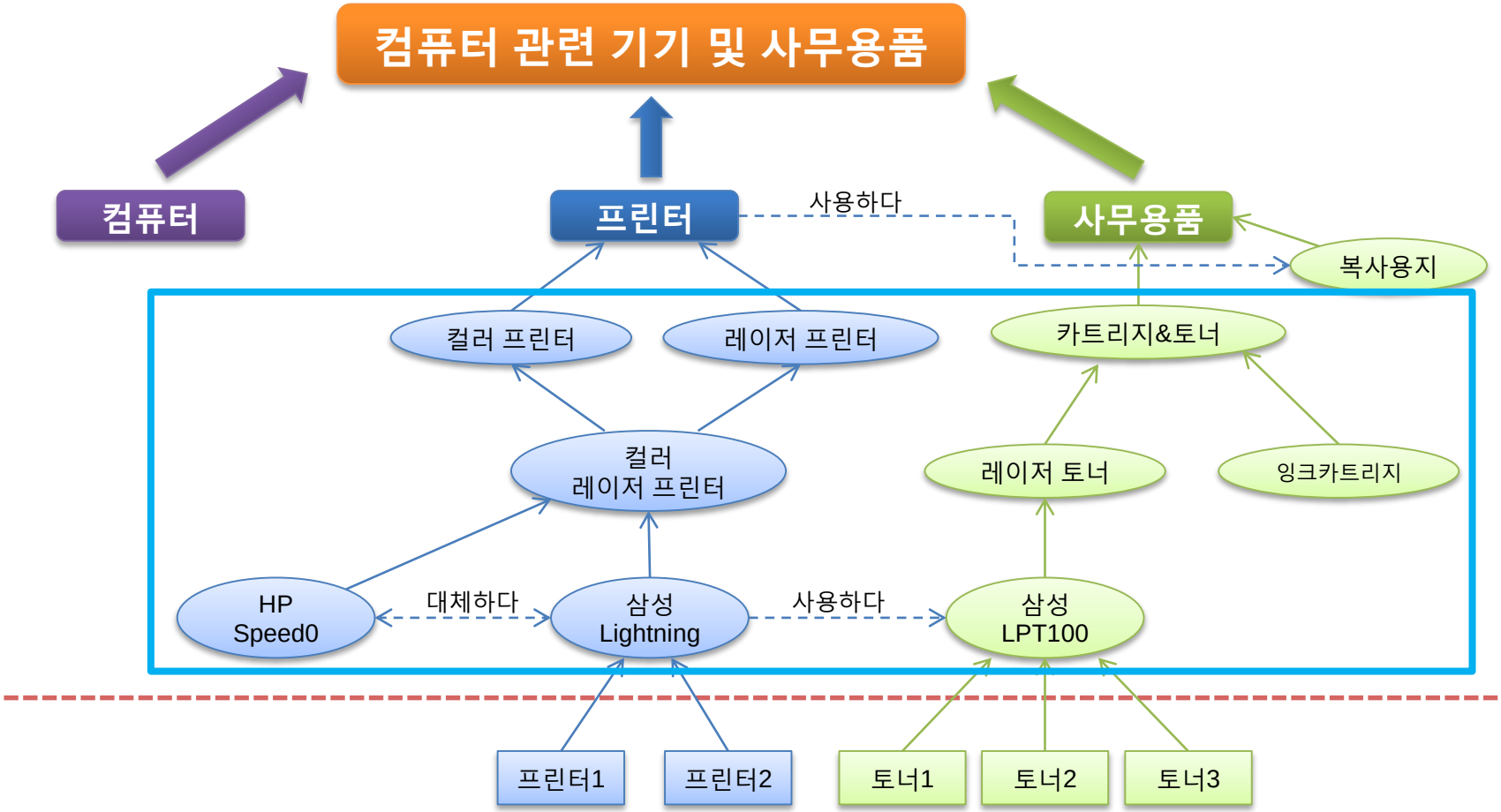
- 모든 올바른(legal) OWL Lite 온톨로지는 올바른 OWL DL 온톨로지이다
- 모든 올바른(legal) OWL DL 온톨로지는 올바른 OWL Full 온톨로지이다
- 모든 타당한(Valid) OWL Lite 결론은 타당한 OWL DL 결론이다
- 모든 타당한(Valid) OWL DL 결론은 타당한 OWL Full 결론이다



# Chapter 7 OWL(Web Ontology Language)

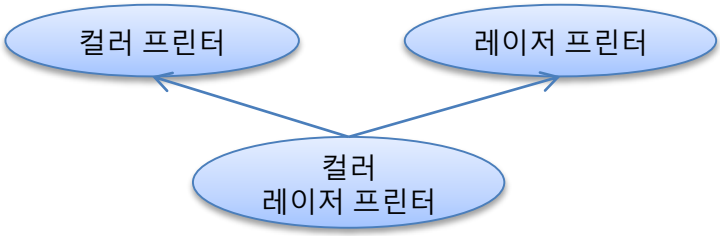
- 1. OWL의 기본 요소 : 클래스와 속성
  - 1.1 클래스, 개체 및 클래스 계층구조
  - 1.2 속성과 계층구조
- 2. OWL의 새로운 기능
  - 2.1 클래스와 속성의 표현
  - 2.2 특별한 속성을 통한 추론 기능의 활성화
  - 2.3 온톨로지 병합과 재사용
- 3. 세 종류의 OWL
  - 3.1 OWL Lite
  - 3.2 OWL DL(Description Logic)
  - 3.3 OWL Full
- 4. OWL 예제

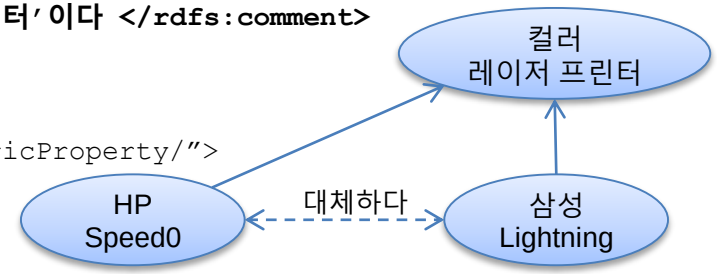
# 4 OWL 예제 (1/4)


컴퓨터 관련 기기 및 사무용품
컴퓨터
프린터
사용하다
사무용품
복사용지
컬러 프린터
레이저 프린터
카트리지&토너
컬러
레이저 프린터
레이저 토너
잉크카트리지
HP
Speed0
대체하다
삼성
Lightning
사용하다
삼성
LPT100
프린터1
프린터2
토너1
토너2
토너3

# 4 OWL 예제 (2/4)

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

<rdfs:comment> 문서 타입 정의 </rdfs:comment>
<!DOCTYPE rdf:RDF[
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-synta6-ns#">
    <!ENTITY owl "=http://www.w3.org/2002/07/owl#">
    <!ENTITY printer "http://Ontolohy.snu.ac.kr/printer#">
]>

<rdfs:comment> 네임스페이스 정의 </rdfs:comment>
<rdf : RDF
    xmlns              ="&printer;"
    xmlns:printer      ="&printer;"
    xmlns:rdf          ="&rdf;"
    xmlns:rdfs         ="&rdfs;"
    xmlns:owl          ="&owl;"
    xmlns:base         ="&printer;">
<owl:Ontology rdf:about=" "/>

<rdfs:comment> 클래스 '컬러 레이저 프린터'는 '컬러 프린터'와 '레이저 프린터'의 하위 클래스이다 </rdfs:comment>
<owl:Class rdf:ID = "컬러레이터프린터">
    <rdfs:subClassOf rdf:resource = "#컬러프린터"/>
    <rdfs:subClassOf rdf:resource = "#레이저프린터"/>
</owl:Class>

<rdfs:comment> 대칭속성 '대체하다'는 정의역과 공역이 모두 '컬러 레이져 프린터'이다 </rdfs:comment>
<owl:ObjectProperty rdf:ID="대체하다">
    <rdfs:domain rdf:resource = "#컬러레이저프린터"/>
    <rdfs:range rdf:resource = "#컬러레이저프린터"/>
    <rdf:type rdf:resource = "http://www.w3.org/2002/07/owl#SymmetricProperty/">
</owl:ObjectProperty>
```

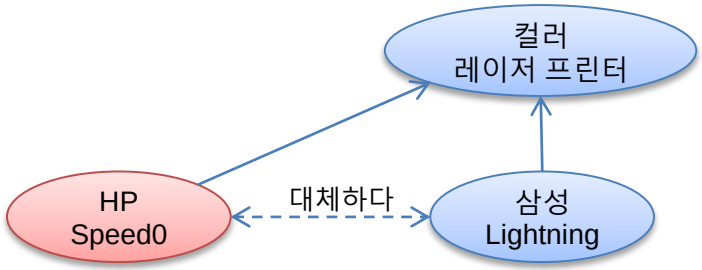


# 4 OWL 예제 (3/4)

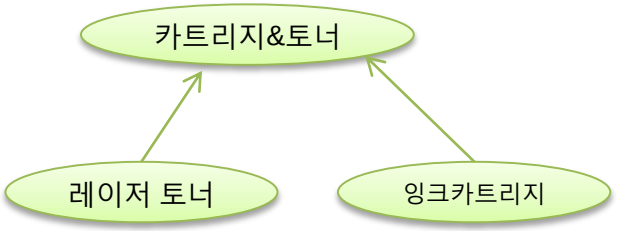
```
<rdfs:comment> 속성 '사용하다'의 정의역은 '프린터' 클래스이고, 공역은 '사무용품' 클래스이다 </rdfs:comment>
<owl:ObjectProperty rdf:ID = "사용하다">
    <rdf:domain rdf:resource = "#프린터"/>
    <rdfs:range rdf:resource = "#사무용품"/>
</owl:ObjectProperty>
```

```
<rdfs:comment> 클래스 'HP Speed0'는 '컬러 레이저 프린터'의 하위 클래스이며 'HP Speed0'프린터는 삼성 Lightning' 프린터만을
대체할 수 있다 </rdfs:comment>
<owl:Class rdf:ID = "HPSpeed0">
    <rdfs:subClassOf rdf:resource="#컬러레이져프린터"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty>
                <owl:Symmetric Property rdf:resource="#대체하다"/>
            </owl:onProperty>
            <owl:allValuesFrom rdf:resource="#삼성Lightning"/>
        </owl:Restriction>
     </rdfs:subClassOf>
</owl:Class>
```



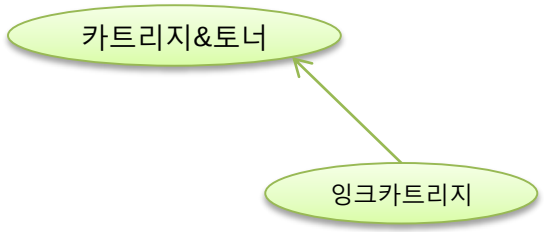

```
<rdfs:comment> '카트리지와 토너' 클래스는 '사무용품' 클래스의 하위 클래스이다 </rdfs:comment>
<owl:Class rdf:ID = "카트리지와토너">
    <rdfs:subClassOf rdf:resource="#사무용품"/>
</owl:Class>
```


```
<rdfs:comment> '레이저 토너'는 '카트리지와 토너'의 하위 클래스이며, '잉크 카트리지'와는 공통의 인스턴스가 없다(비접합성을 지닌다)
</rdfs:comment>
<owl:Class rdf:ID = "레이저 토너">
    <rdfs:subClassOf rdf:resource = "#카트리지와토너"/>
    <owl:disjointWith rdf:resource = "#잉크카트리지"/>
</owl:Class>
```



# 4 OWL 예제 (4/4)

```
<rdfs:comment> '잉크 카트리지'는 '카트리지와 토너'의 하위 클래스이다 </rdfs:comment>
<owl:Class rdf:ID = "잉크카트리지">
    <rdfs:subClassOf rdf:resource="#카트리지와 토너"/>
</owl:Class>
```




```
<rdfs:comment> '삼성'Lightning'은 '컬러 레이저 프린터'의 하위 클래스이며, '삼성 Lightning'프린터는 '삼성 LPT100' 만을 사용
한다. '삼성 LPT100'은 '레이저 토너'의 하위 클래스이다.</rdfs:comment>
<owl:Class rdf:ID = "삼성 Lightning">
    <rdfs:subClassOf rdf:resource = "#컬러레이저프린터"/>
    <rdfs;subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource = "#사용하다"/>
            <owl:allValuesFrom>
                <owl:Class rdf:ID = "삼성LPT100">
                    <rdfs:subClassOf rdf:resource="#레이저토너"/>
                </owl:Class>
            </owl:allValuesFrom>
        </owl:Restriction>
     </rdfs:subClassOf>
</owl:Class>

</rdf:RDF>
```