

子集判断

2011.8

题 目

请写一个算法，判断一个查询表达式是不是另外一个查询表达式的子集。

稍微加一点解释，就拿SQL语句为例，如果查询A是 `age > 21` 查询B是 `age > 20` 我们知道，A所代表的集合是B所代表的集合的子集。凡是满足A的条件的记录，也一定满足B的条件。我需要一个算法，在给出任意两个表达式以后，判断一个是不是另外一个是子集。

```
if ($queryA->isSubSet($queryB)) { echo("A is subset of B");}
```

为了简单起见，只需要实现最简单的AND, OR逻辑操作，大于，等于，小于三种比较操作就好。最好用PHP，其他语言，比如Java也没问题。

这有什么用呢？

这是个我们不到10人的技术团队日常典型需要解决的问题。给个例子：百姓网为了应付更大的搜索数据量，把搜索分布在多个城市的多台服务器上。系统管理员可以根据数据的使用频度等规律，配置几个不同的数据库（MySQL的，和Solr的）。这样，当有一个新的广告出来后，子集算法根据搜索库的配置查询就可以决定，它更新到哪一个或多个库里面。

查询的时候，如果确认给定的查询条件是以前配置的一个库的子集，就可以只从那个库里查询了。这可以让我们轻松地配置几十个搜索库，而不用改一行代码。

<http://blog.baixing.com/?p=9>

解題思路

最简单的表达式比较

$A: a > 10$

$B: a > 15$

is A subset of B ?

- 表达式由 字段名 比较操作符 数值 三部分组成
- A 和 B 的字段名相同是“ A 是 B 的子集”的必要条件
- 还需要根据 A 和 B 的比较操作符与数值判断

A: $a > 10$

B: $a > 15$

is *A* subset of *B*?

- **an='a'**, **ao='>'**, **av=10**
bn='a', **bo='>'**, **bv=15**
- 当 **ao='>'** 且 **bo='>'** 时, 如果 '**bv>=av**' 则 **A**是**B**的子集
- 比较操作符: **> < = >= <= !=**
一共有 6×6 种组合情况

判断最简单的表达式只需要列出所有36种情况即可

简单表达式与 简单的复合表达式比较

$A: a > 10 \text{ and } b > 10$

$B: a > 15$

is A subset of B ?

- 简单的复合表达式由 简单表达式 逻辑操作符 简单表达式 三部分组成
- 将简单表达式称为 factor 用 X 、 Y 、 Z 表示

is $X \subseteq (Y \cap Z)$?

is $X \subseteq (Y \cup Z)$?

is $(X \cap Y) \subseteq Z$?

is $(X \cup Y) \subseteq Z$?

X 、 Y 、 Z 都是简单表达式

- is $X \subseteq (Y \cap Z)$? $\Rightarrow X \subseteq Y$ 且 $X \subseteq Z$
- is $X \subseteq (Y \cup Z)$? $\Rightarrow X \subseteq Y$ 或 $X \subseteq Z$
- is $(X \cap Y) \subseteq Z$? $\Rightarrow X \subseteq Z$ 或 $Y \subseteq Z$
- is $(X \cup Y) \subseteq Z$? $\Rightarrow X \subseteq Z$ 且 $Y \subseteq Z$

简单表达式与全OR连接的复合表达式比较

$A: a > 10$

$B: a > 15 \text{ or } a > 5 \text{ or } b > 10$

is A subset of B ?

is $X \subseteq (I \cap J \cap K)$?

is $(I \cap J \cap K) \subseteq X$?

is $X \subseteq (I \cap J \cap K) \Rightarrow X \subseteq I \text{ 且 } X \subseteq J \text{ 且 } X \subseteq K$

is $(I \cap J \cap K) \subseteq X \Rightarrow I \subseteq X \text{ 或 } J \subseteq X \text{ 或 } K \subseteq X$

简单表达式与全AND连接 的复合表达式比较

$A: a > 10$

$B: a > 15 \text{ and } a > 5 \text{ and } b > 10$

is A subset of B ?

is $X \subseteq (I \cup J \cup K)$?

is $(I \cup J \cup K) \subseteq X$?

is $X \subseteq (I \cup J \cup K) \Rightarrow X \subseteq I \text{ 或 } X \subseteq J \text{ 或 } X \subseteq K$

is $(I \cup J \cup K) \subseteq X \Rightarrow I \subseteq X \text{ 且 } J \subseteq X \text{ 且 } K \subseteq X$

复合表达式比较

A: $a > 10 \text{ and } a < 20$

B: $a > 15 \text{ and } a > 5 \text{ or } a < 25$

is A subset of B?

将表达式标准化为全以OR相连的表达式；OR里的子表达式为简单表达式或全以AND连接的表达式

A: $(a > 10 \text{ and } a < 20)$

B: $(a > 15 \text{ and } a > 5)$

OR

$(a < 25)$

表达式标准化

($a > 10$ or $a < 20$) and ($b > 10$ or $b < 20$)

----->

($a > 10$) AND ($b > 10$ or $b < 20$)

OR

($a < 20$) AND ($b > 10$ or $b < 20$)

----->

($a > 10$ AND $b > 10$)

OR

($a > 10$ AND $b < 20$)

OR

($a < 20$ AND $b > 10$)

OR

($a < 20$ AND $b < 20$)

复合表达式比较

A: $(a > 10 \text{ or } a < 20) \text{ and } (b > 10 \text{ or } b < 20)$

B: $(a > 5 \text{ or } a < 25) \text{ and } (b > 5 \text{ or } b < 25)$

is *A* subset of *B*?

($a > 10 \text{ AND } b > 10$)
OR
($a > 10 \text{ AND } b < 20$)
OR
($a < 20 \text{ AND } b > 10$)
OR
($a < 20 \text{ AND } b < 20$)

subset of

($a > 5 \text{ AND } b > 5$)
OR
($a > 5 \text{ AND } b < 25$)
OR
($a < 25 \text{ AND } b > 5$)
OR
($a < 25 \text{ AND } b < 25$)

复合表达式比较

(a > 10 AND b > 10)	subset of	(a > 5 AND b > 5)
OR		OR
(a > 10 AND b < 20)		(a > 5 AND b < 25)
is		OR
OR		(a < 25 AND b > 5)
(a < 20 AND b > 10)		OR
OR		(a < 25 AND b < 25)
(a < 20 AND b < 20)		

is $(X \cup Y) \subseteq (I \cup J)$? $\Rightarrow X \subseteq (I \cup J)$ 且 $Y \subseteq (I \cup J)$

is $(X \cap Y) \subseteq (I \cap J)$? $\Rightarrow (X \cap Y) \subseteq I$ 且 $(X \cap Y) \subseteq J$

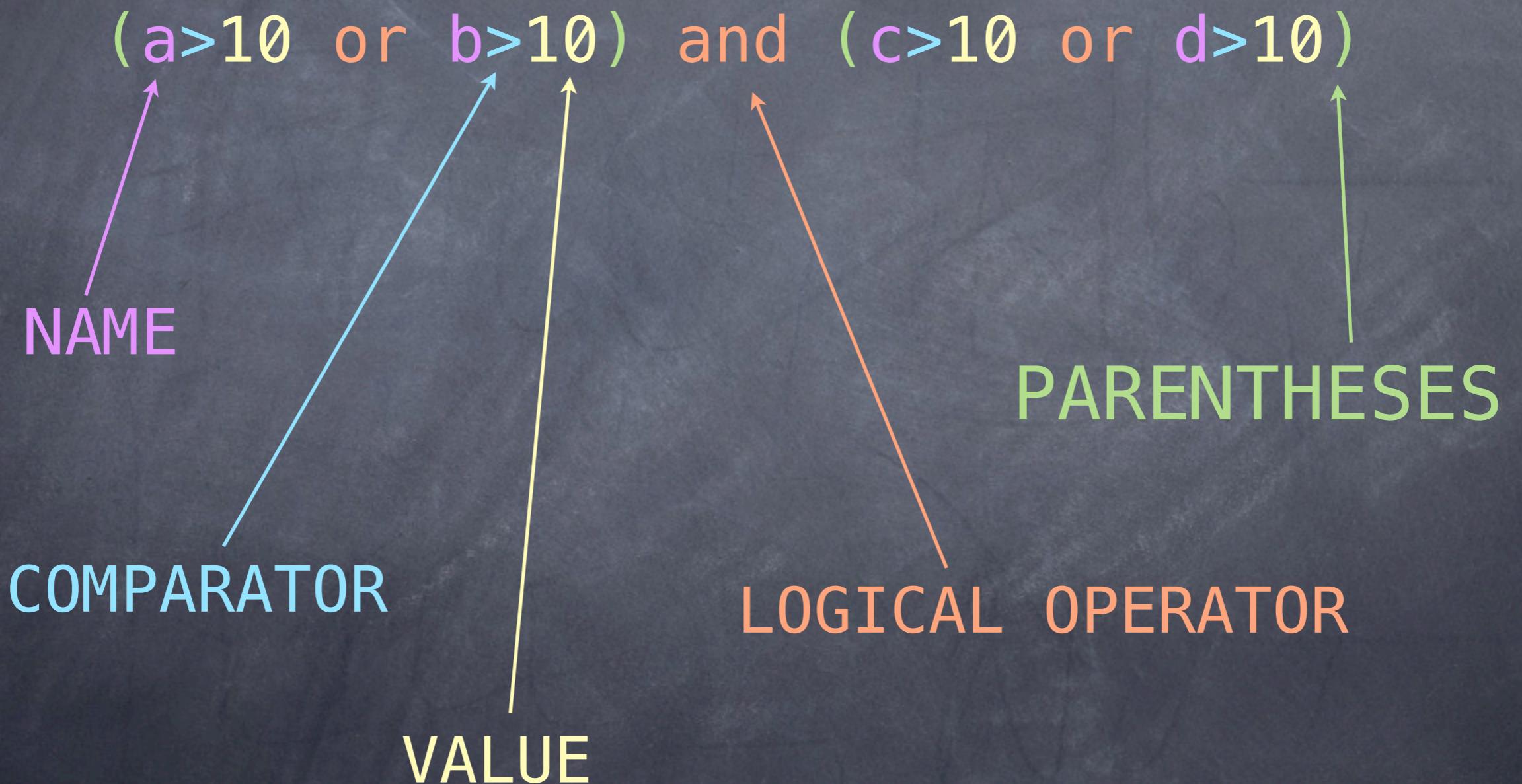
X、Y、I、J是简单表达式或全AND连接的表达式

实现

表达式的组成

(a>10 or b>10) and (c>10 or d>10)

表达式的组成



最简单的表达式

- age>20
- NAME COMPARATOR VALUE
- 把这种最简单的表达式称为 factor

表达式语法规则

```
expr   : factor
      | NOT expr
      | expr AND expr
      | expr OR expr
      | LPAREN expr RPAREN
```

```
factor : NAME COMPARATOR value
```

lex and Yacc

- 表达式的数据结构伪代码为

```
typedef struct expr_t {  
    struct expr_t *left;  
    operator_t      op;  
    struct expr_t *right;  
}
```

```
def is_subset(a, b):
    ....
    if ao == '||':
        return is_subset(al, b) and is_subset(ar, b)
    elif bo == '||':
        return is_subset(a, bl) or is_subset(a, br)
    elif bo == '&&':
        return is_subset(a, bl) and is_subset(a, br)
    elif ao == '&&':
        return is_subset(al, b) or is_subset(ar, b)
    else:
        return is_subset_factor(a, b)
```

这里 if 的顺序很重要

```
def is_subset_factor(a, b):
    ...
    if an != bn:
        return False

    if ao == '==':
        if bo == '==' : return bv == av
        if bo == '>' : return bv < av
        if bo == '<' : return bv > av
        if bo == '>=': return bv <= av
        if bo == '<=': return bv >= av
        if bo == '!=': return bv != av

    elif ao == '>':
        if bo == '==' : return False
        if bo == '>' : return bv <= av
        if bo == '<' : return False
        if bo == '>=': return bv <= av
        if bo == '<=': return False
        if bo == '!=': return bv <= av

    ...

```

单元测试

- `test_factor()` 可以覆盖所有情况

```
(False, 'a>10', 'a=10'),      (False, 'a>10', 'a=5'),      (False, 'a>10', 'a=15'),  
(True, 'a>10', 'a>10'),       (True, 'a>10', 'a>5'),       (False, 'a>10', 'a>15'),  
(False, 'a>10', 'a<10'),       (False, 'a>10', 'a<5'),       (False, 'a>10', 'a<15'),  
(True, 'a>10', 'a>=10'),      (True, 'a>10', 'a>=5'),      (False, 'a>10', 'a>=15'),  
(False, 'a>10', 'a<=10'),      (False, 'a>10', 'a<=5'),      (False, 'a>10', 'a<=15'),  
(True, 'a>10', 'a!=10'),       (True, 'a>10', 'a!=5'),       (False, 'a>10', 'a!=15'),
```

- 其他测试可以覆盖典型情况