My Own “Parallel Parser” for ”Paradox Game Data”

DongWook Lee
vztpv@naver.com

발표자 소개

Working from home

DongWook Lee

vztpv

Edit profile

Interested in C++, Parsing, Parallel Processing, Simulation.

South Korea

vztpv@naver.com

Overview | Repositories 77 | Projects 0 | Stars 113 | Followers 11 | Following 30

Pinned

Customize your pins

ClauText/ClauText

My Own Language for Editing? Paradox Game(Clausewitz Engine) Data , version 1.1.7~

C++ ★ 2

ClauText/ClauToCpp

ClauText To Cpp - Using ClauText 1.1.6

C++

ClauText/ClauExplorer

ClauExplorer with ClauText version 1.1.7~

C++

EU3ToVIC2

EU3 to VIC2 Using Idhrendur/paradoxGameConverters and My ClauTextAPI?

C++ ★ 1

ClauText/ClauLint

Next Project for ClauText, ClauText Schema Maker? and Validator (check JsonLint) with ClauText 1.1.3

C++

ClauText/ClauParser

My Own "Parallel Parser" for "Paradox Game Data" (clausewitz engine or jomini engine)

C++

목차

- ParadoxGame and ParadoxGameData
- Parallel Lexing(Scanning)
- Parallel Parsing
- Application

Paradox Game?

Paradox Game?

여기서는 Paradox Interactive 회사에서 클라우제비츠(clausewitz), jomini 엔진으로 **만든 게임**을 의미하며,

Paradox Parser는 이러한 게임 엔진으로 만든 게임의 텍스트 파일의 주된 스타일을 파싱합니다

Jomini Engine

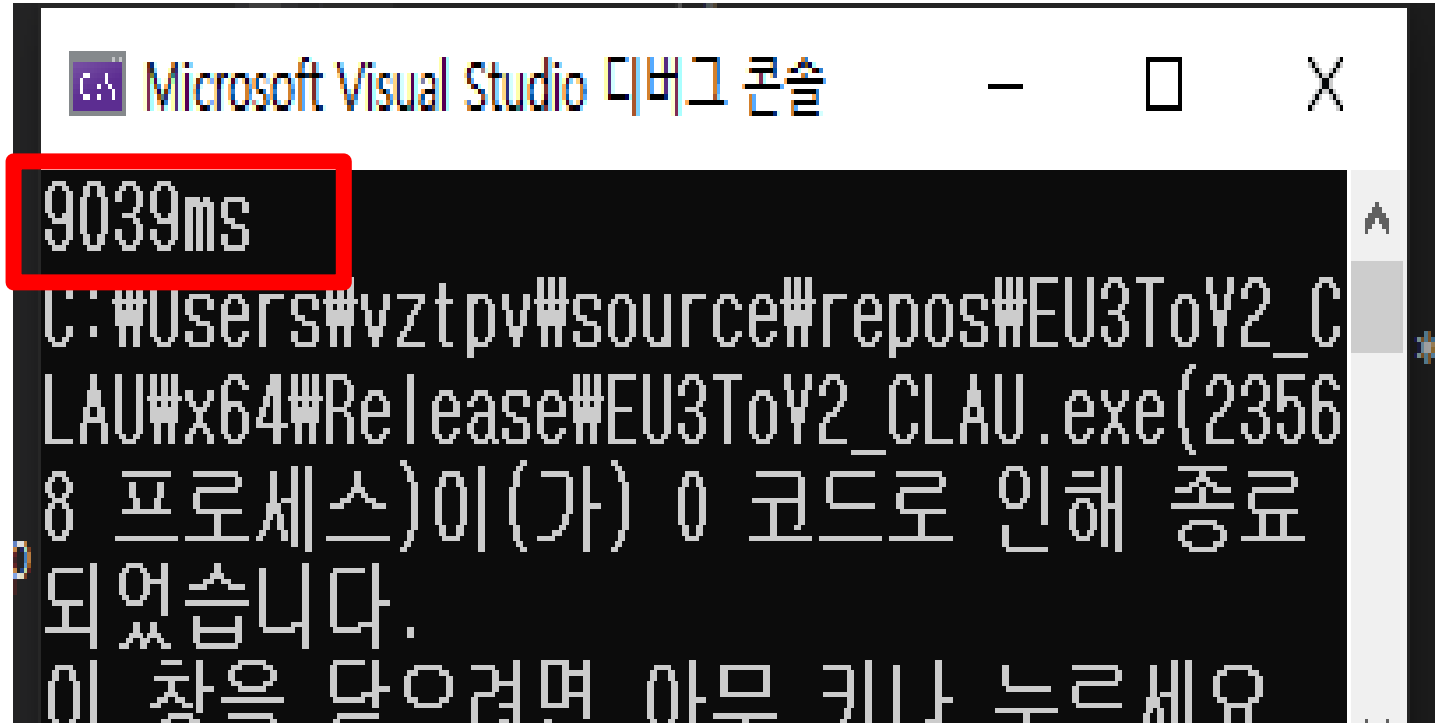
Clausewitz Engine

유로파 유니버셜리스4 (EU4)

- 역사 전략 시뮬레이션 게임
- 1444년 ~ 1821년 배경
- 매우 복잡함 (군사, 경제, 정치, 종교 등등)
- 진입장벽이 높음
- 세이브파일의 용량이 큼(30~60MB, 2백만줄)
- 다른 Paradox 게임도 세이브파일의 용량이 큽니다

Other Parsers

- Most Parsers are Very Slow for Large(>30MB) File
- A Parser used in **Idhrendur/paradoxGameConverters** is Faster than most other parsers.
- (test for 50MB)



My Grammar for “Paradox Game Data”

- S -> A
- A -> var = val A | var = { A } A | { A } A | val A | λ
- var, val : 문자열!

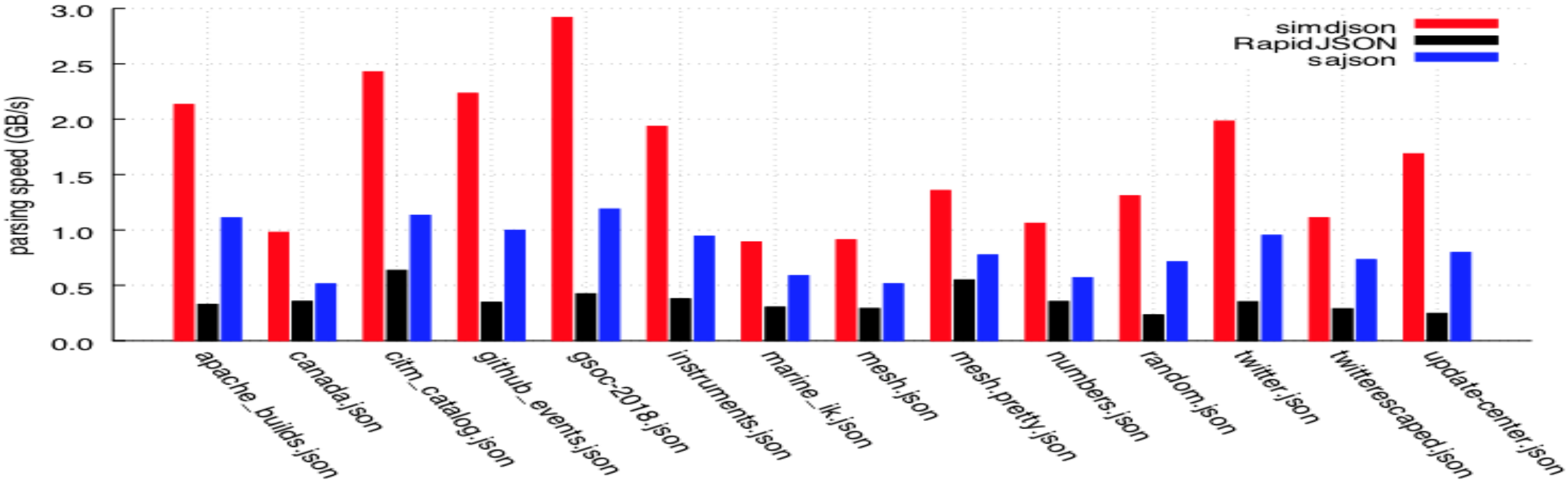
“Paradox Game Data” vs JSON

```
# line comment
date=1947.5.8
save_game="Russia.eu4"
savegame_versions={
        "1.18.1.0"
        "1.18.2.0"
        "1.18.3.0"
        "1.18.4.0"
        "1.19.1.0"
}
```

```
{
  “date” : “1947.5.8”,
  “save_game”:"Russia.eu4",
   “savegame_versions”:[
        "1.18.1.0",
        "1.18.2.0",
        "1.18.3.0",
        "1.18.4.0",
        "1.19.1.0"
  ]
}
```

Other Parsers - Json Parsers

- RapidJson - Sequential?, one of the fastest parsers
- SimdJson - Parallel, 2~3 Times Faster than RapidJson



My Rule!

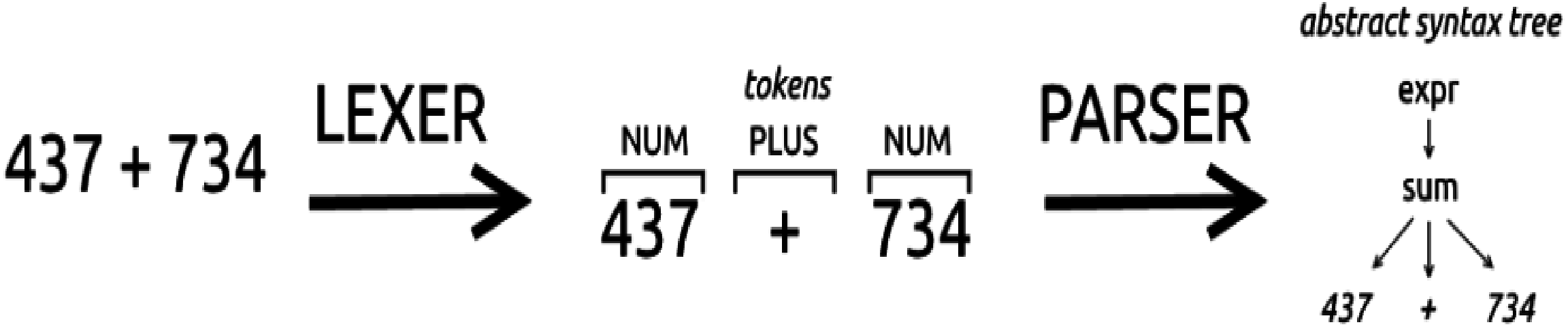
- No Boost
- No Regualr Expression Library
- No std::string`s substr method
- Postpone use of std::string

주의사항

- 데이터 순서가 중요함 - std::vector vs std::map
- 주석은 무시됨(in scanning)
- 중복된 key가 들어 올 수 있음

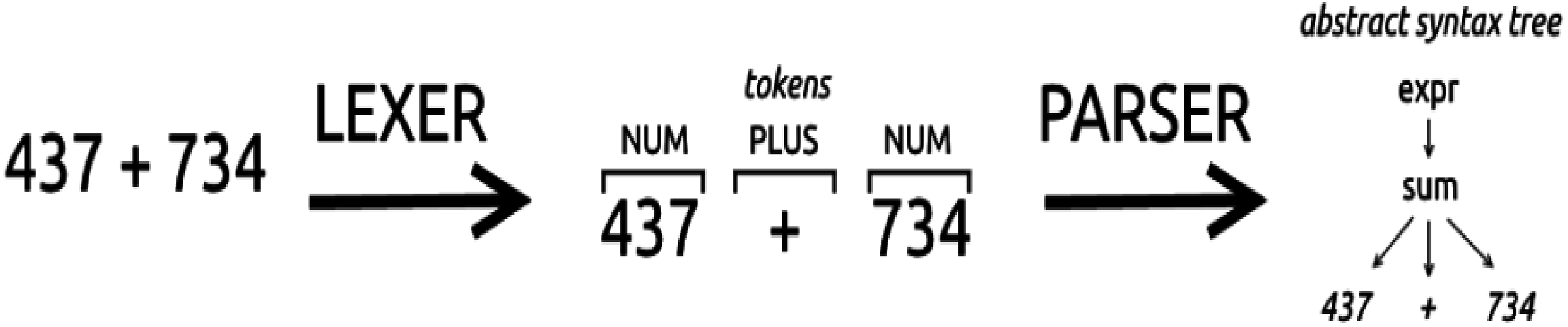
Parsing?

- String -> Tokens -> Tree



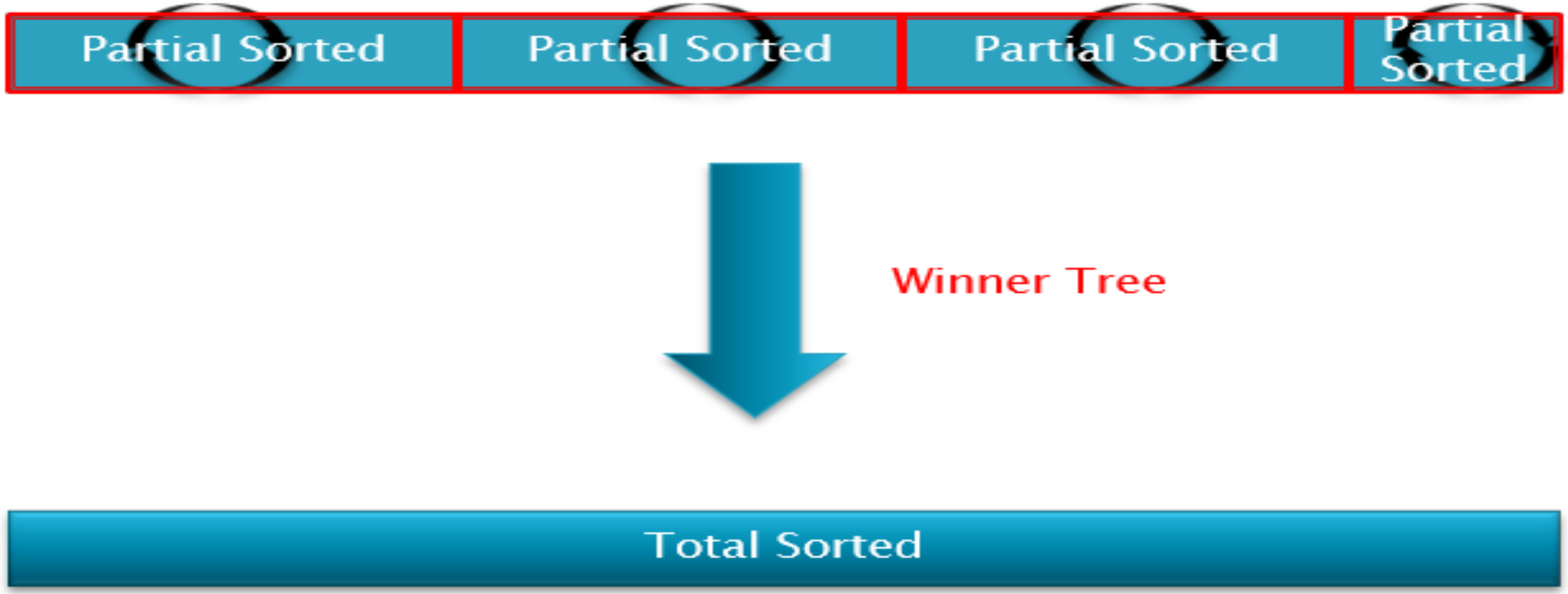
"Parallel Parser"?

- Parallel Scanning(Lexing)
- Parallel Parsing
- using std::thread

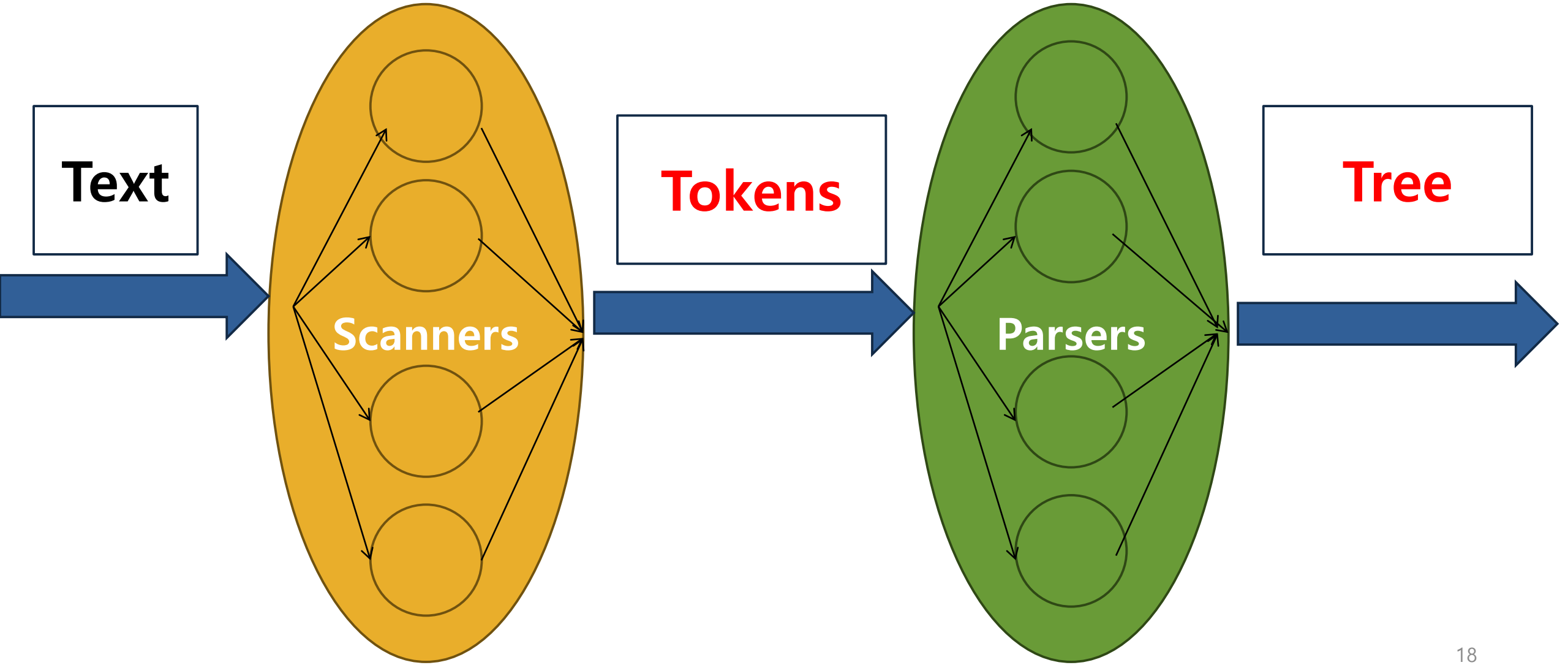


Parallel Sorting?

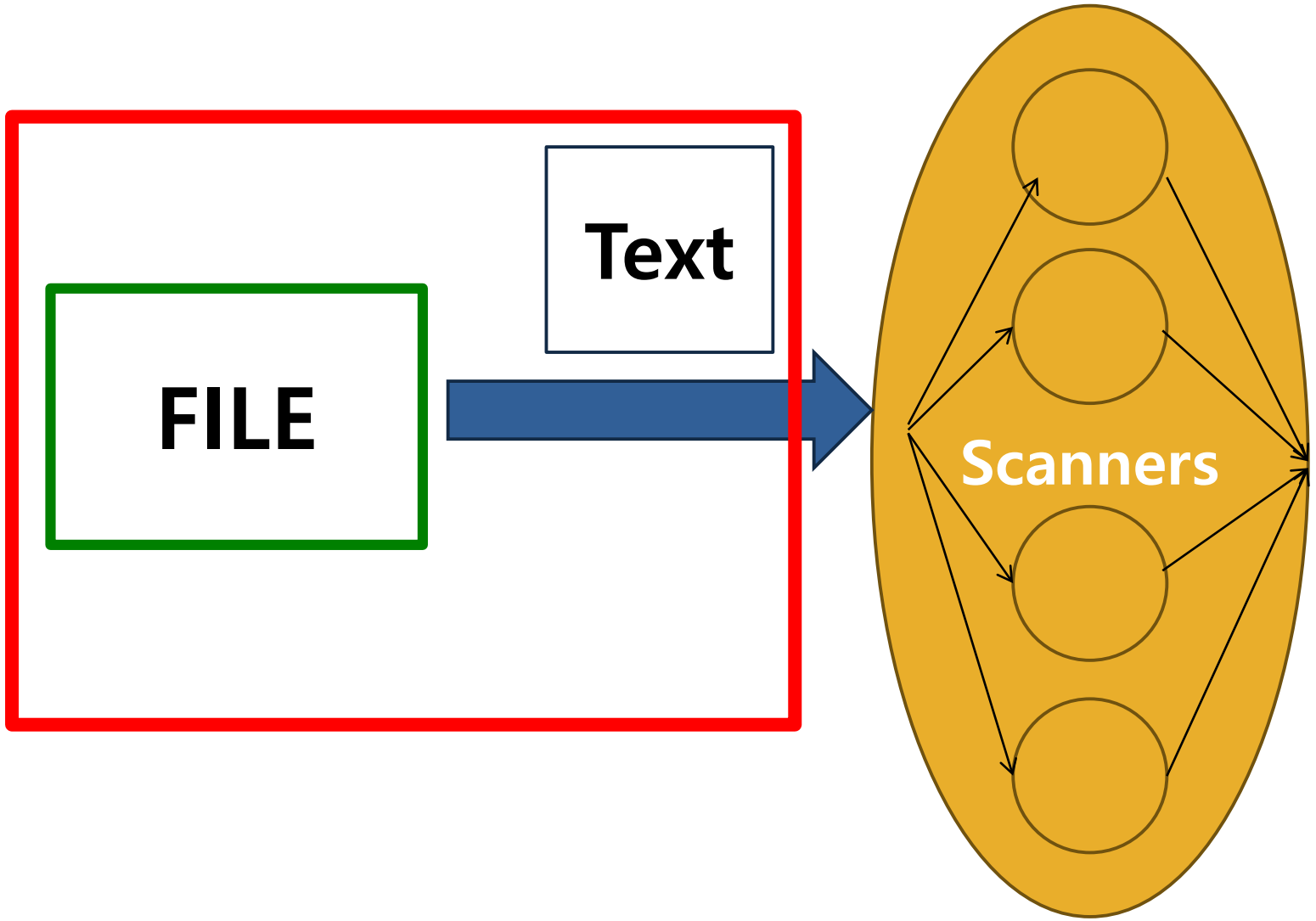
- **Divide, Partial Sorting, Merge**



전체 과정

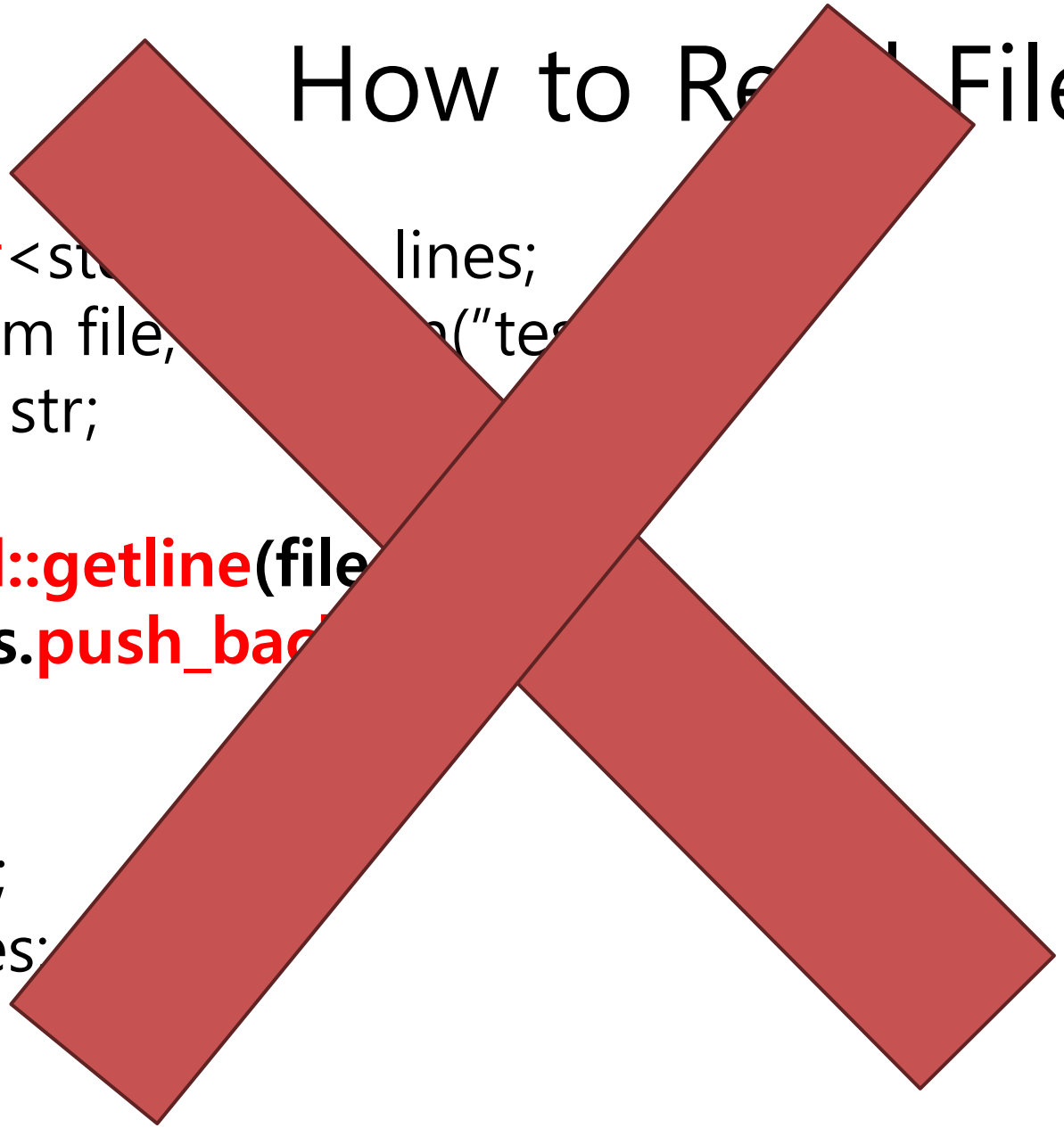


Loading Text From File



How to Re File?

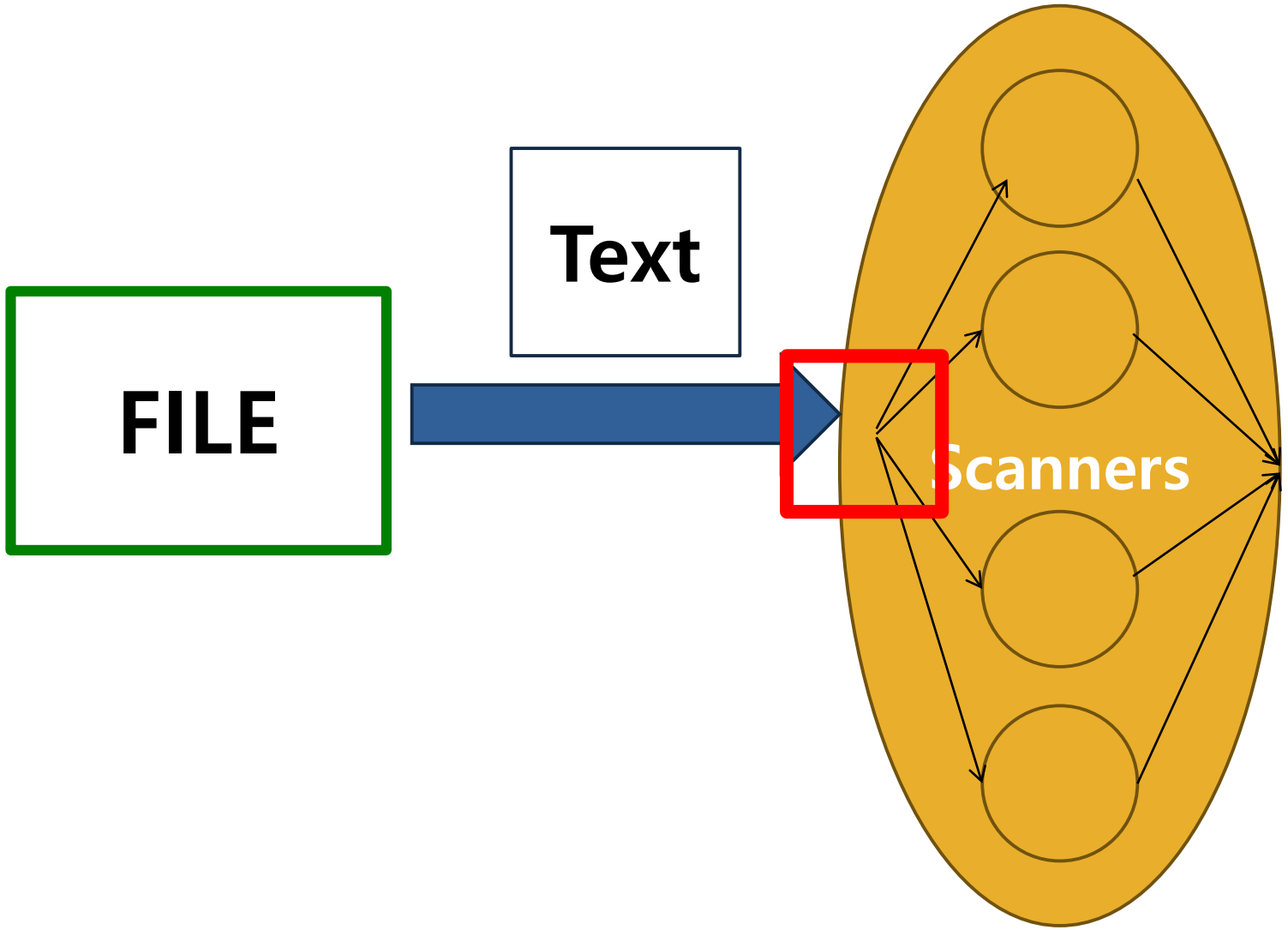
```
std::vector<st                lines;
std::ifstream file,           ("te
std::string str;

while (std::getline(file
        lines.push_bac
}

file.close();
return lines;
```



How to Read File!

```
char* buffer;
std::ifstream file; file.open("test.txt", std::ios::binary);
long long buffer_len = get_file_size(file);

buffer = new char[buffer_len + 1];

file.read(buffer, buffer_len); buffer[buffer_len] = '₩0';

file.close();
return buffer;
```

Convert Text to Tokens



How to Divide Text?

- “ **test = { } ”**
- # test2 = { }
- 엔터키('₩n')에 주목하자

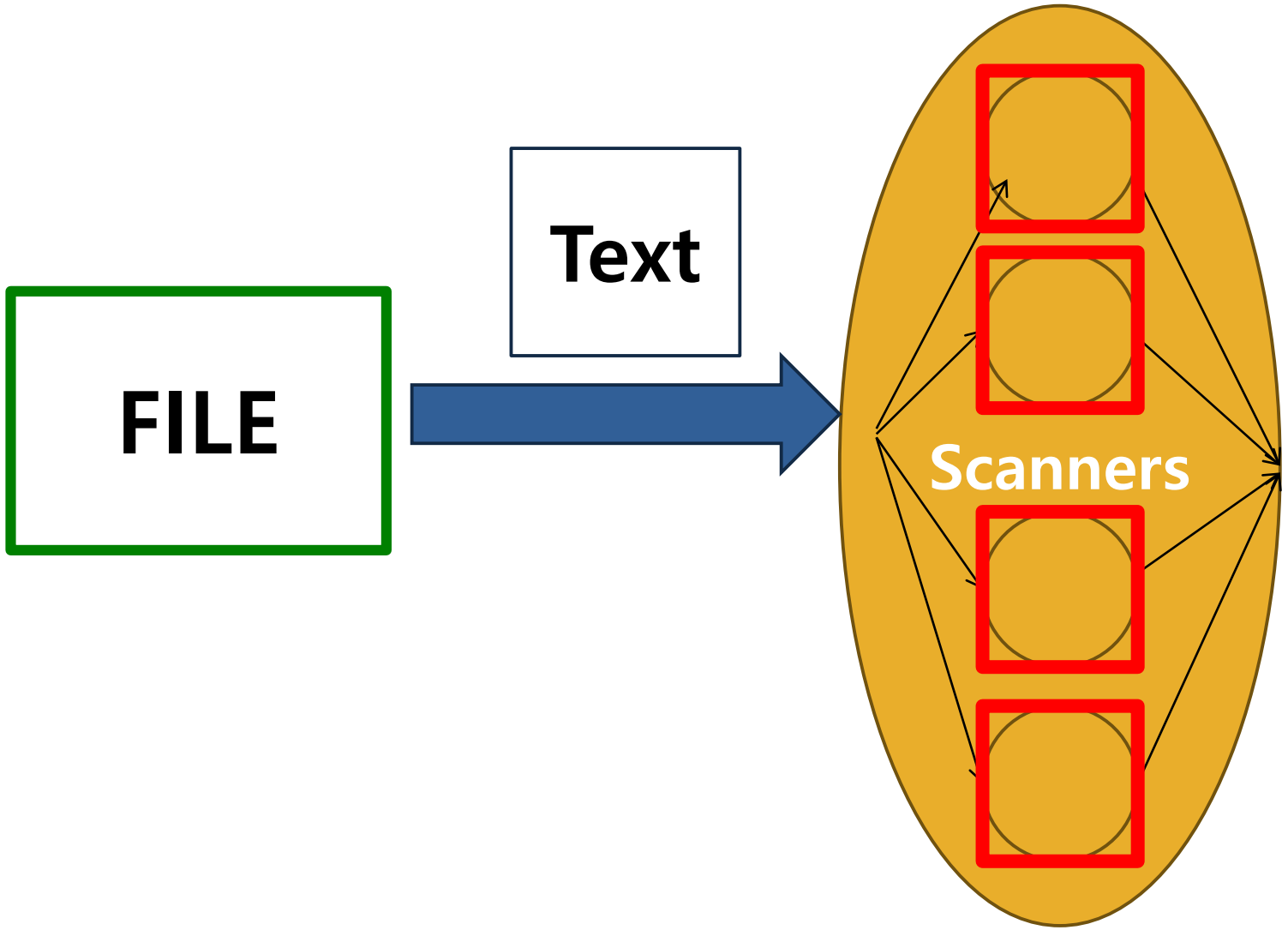
Parallel Scanning - Assumption

- “ “ quoted string must not have ‘₩n’
- “ abc ₩n def ₩n ghi “ # supported
- **“ abc**
  def
  ghi “ # not supported

Parallel Scanning - Precautions

- line comment(#) is not saved
- data is divided by '₩n'

Convert Text to Tokens



Scanning

- 따옴표로 둘러싸인 문자열 처리!
- start **state** : 0
- **state** == 0 and now char == '₩"' : **state** = 1
- **state** == 1 and now char == '₩"' : **state** = 0
- cf) "test ₩" test ₩" "

- and parser support ₩", but not support ₩₩ , ₩₩"

Revisit SimdJson?

```
{ "\\\"Nam[{": [ 116,"\\\\" , 234, "true", false ], "t":"\\\"" }: input data
__1___1____1______1___1______1___1__________1_1_1___11__: Q
_____1_____________________________________________________1___: OD
__1_______1______1___1______1___1__________1_1_1___1__: Q &= ~OD
__111111111_______11111_______11111__________11__11111___: CLMUL(Q,~0)
```

Fig. 3: Branchless code sequence to identify quoted range excluding the final quote character. CLMUL refers to the carry-less multiplication. We assume that OD is computed using the code sequence from Fig. 2.

```
______1_____________________________________________________1___: OCE = OC & E
______1_____________________________________________________1___: OD2 = OCE & E
// merge results, yielding ends of all odd-length sequence of backslashes
______1_____________________________________________________1___: OD = OD1 | OD2
```

How to Tokenize?

```
class Token {
        std::string str;
        long type;
};
```

```
std::vector<std::string> lines;
std::ifstream file; file.open("test.txt");
std::string str;

while (std::getline(file, str)) {
        lines.push_back(str);
}

file.close();
return lines;
```

Token token = **getToken**();

in getToken function, maybe **substr** is used…

How to Tokenize?

char* buffer = readFile("filename.txt");

long long token;

// 32 bit : start index of token in buffer

// 30 bit : length of token

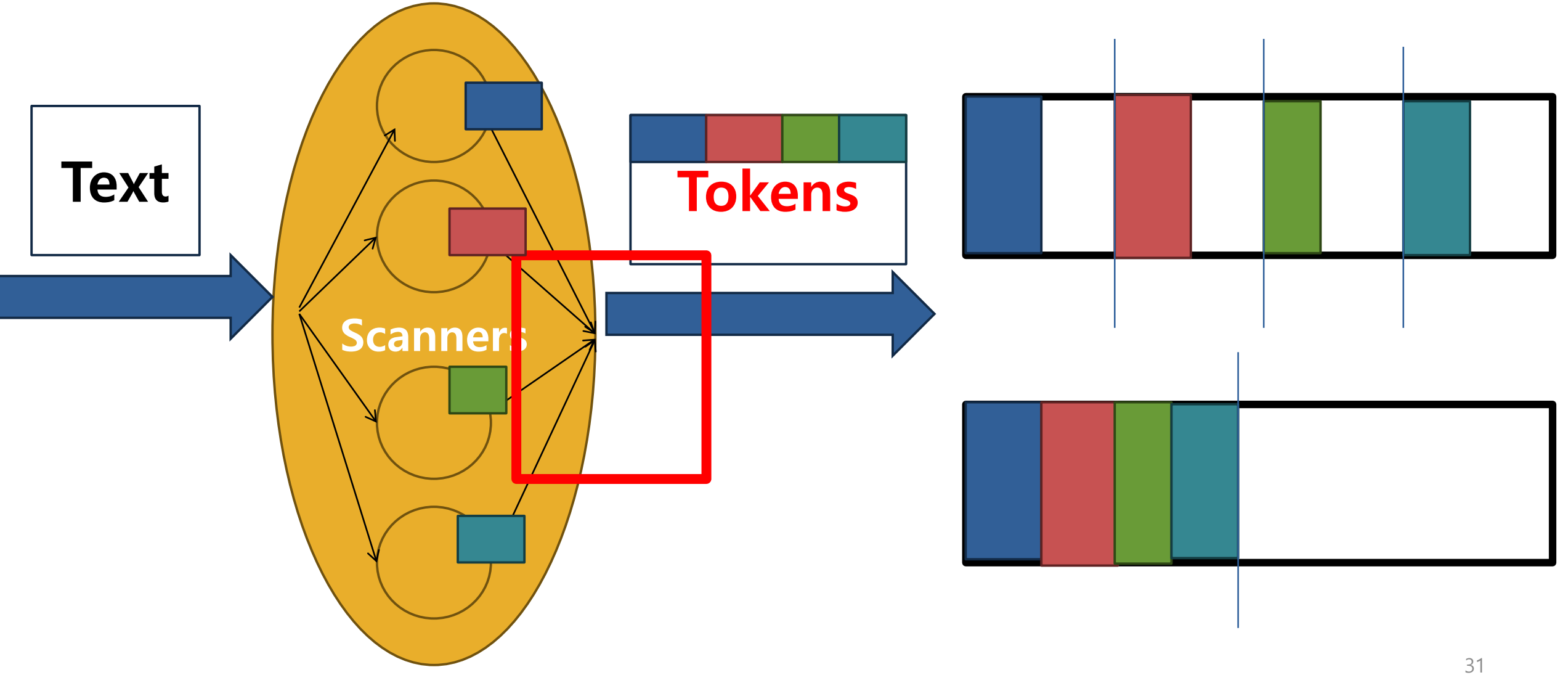
// 2 bit : type of token

(0 - **general**, 1 - **{**, 2 - **}**, 3 - **=**)

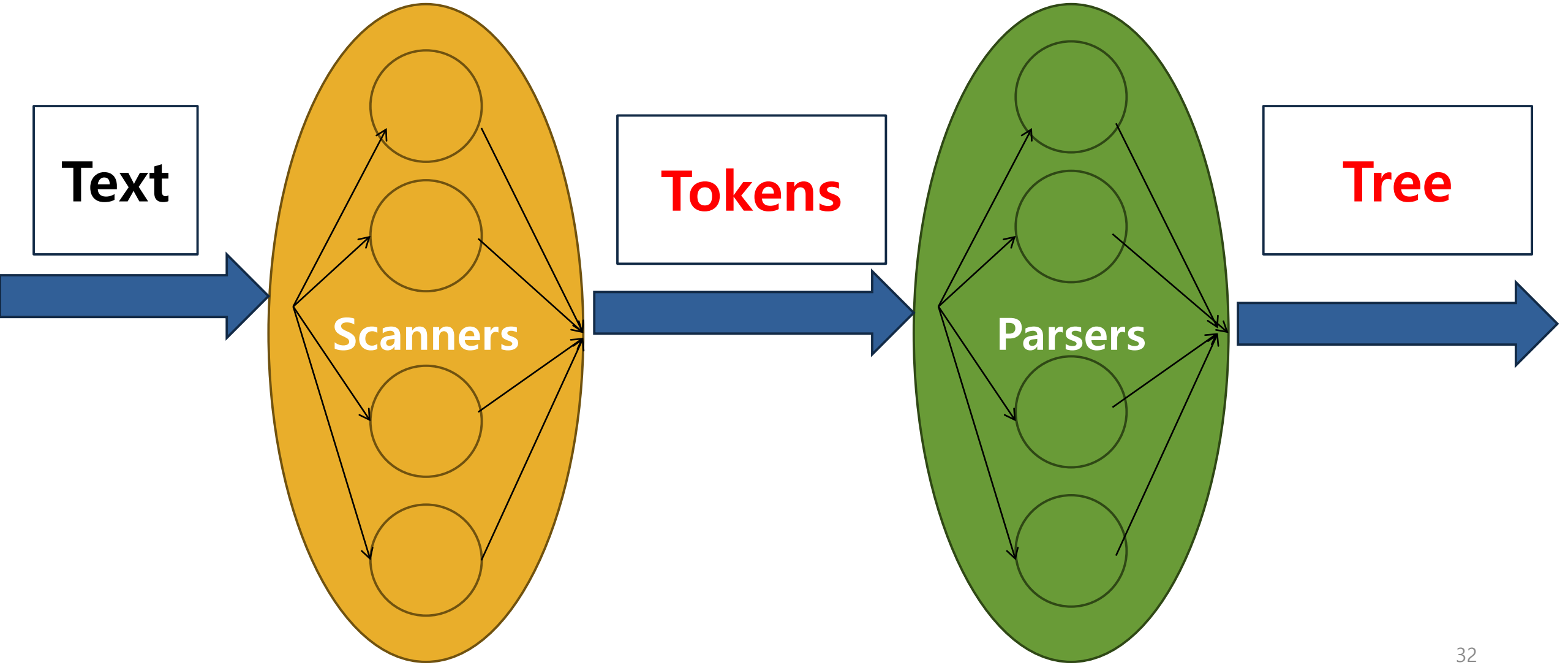
```
char* buffer;
std::ifstream file; file.open("test.txt", std::ios::binary);
long long buffer_len = get_file_size(file);
buffer = new char[buffer_len + 1];
file.read(buffer, buffer_len); buffer[buffer_len = '₩0';
file.close();
return buffer;
```

cf) std::string(buffer + GetIdx(token), GetLength(token));

Merging Tokens

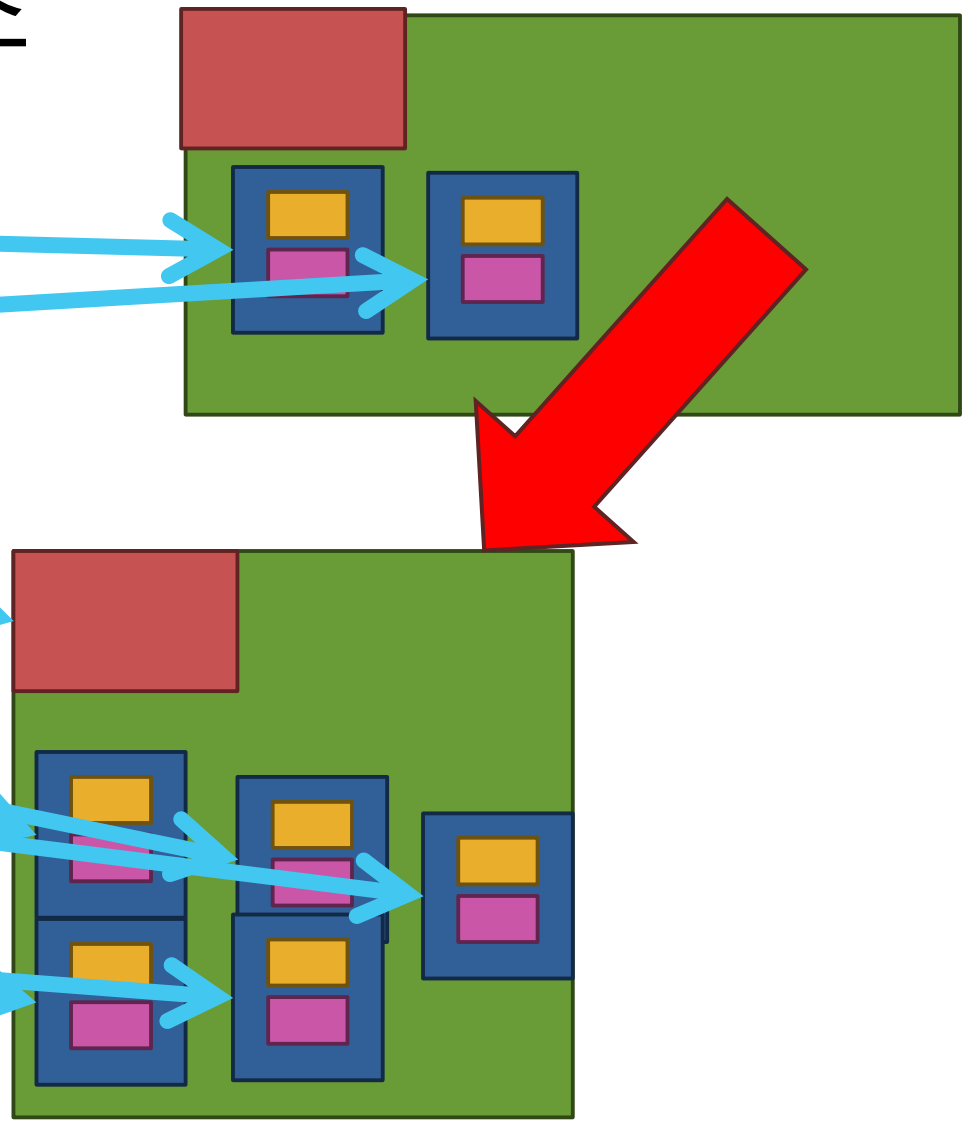


전체 과정

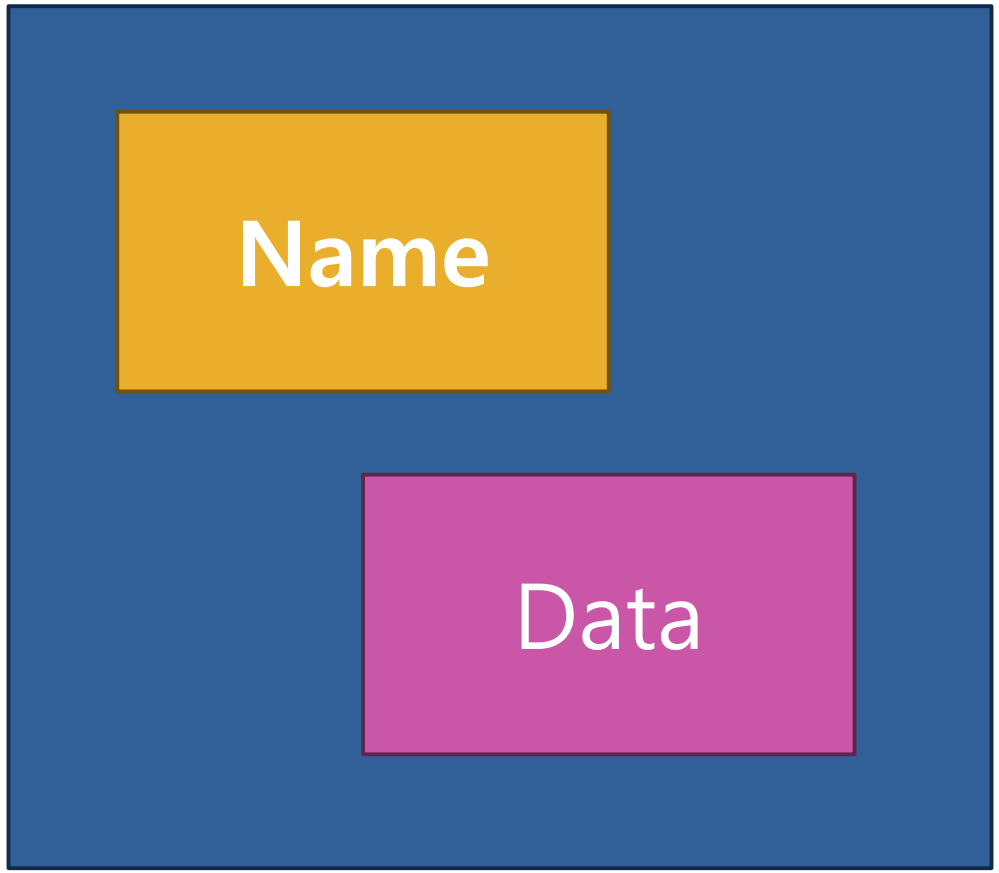


Tree구조

```
date=1947.5.8
save_game="Russia.eu4"
savegame_versions={
        "1.18.1.0"
        "1.18.2.0"
        "1.18.3.0"
        "1.18.4.0"
        "1.19.1.0"
}
```

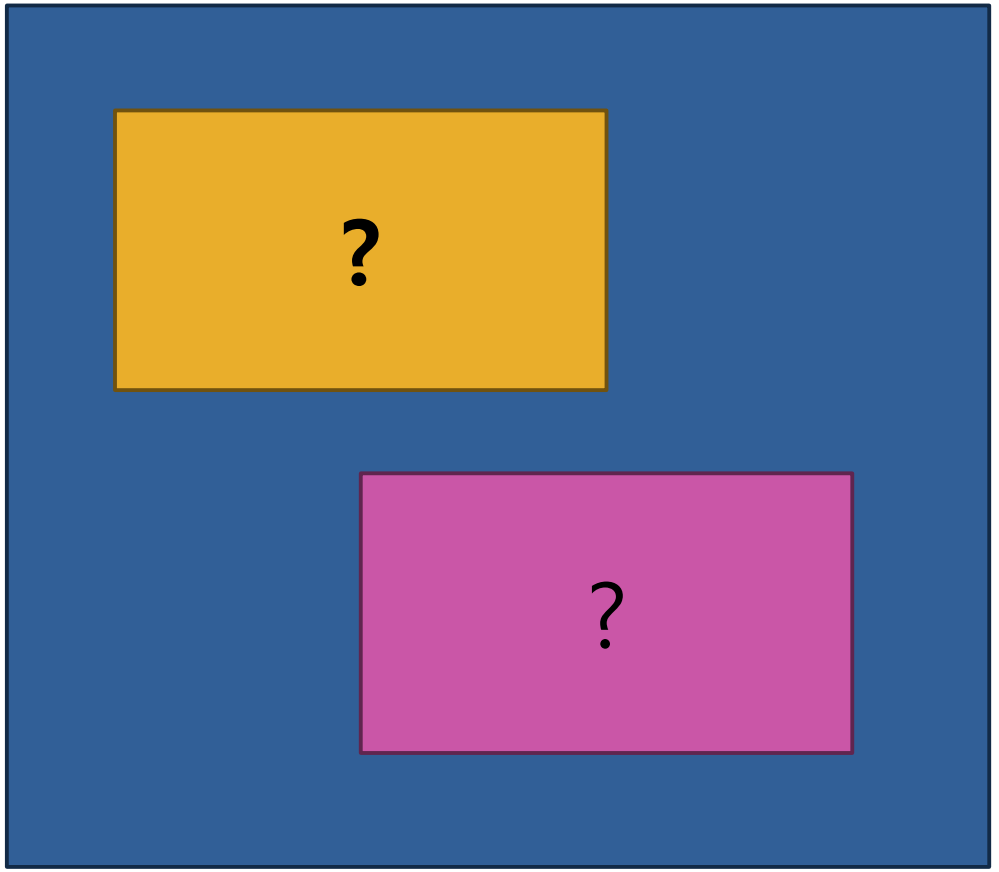


ItemType



ItemType

date = 2019.04.13

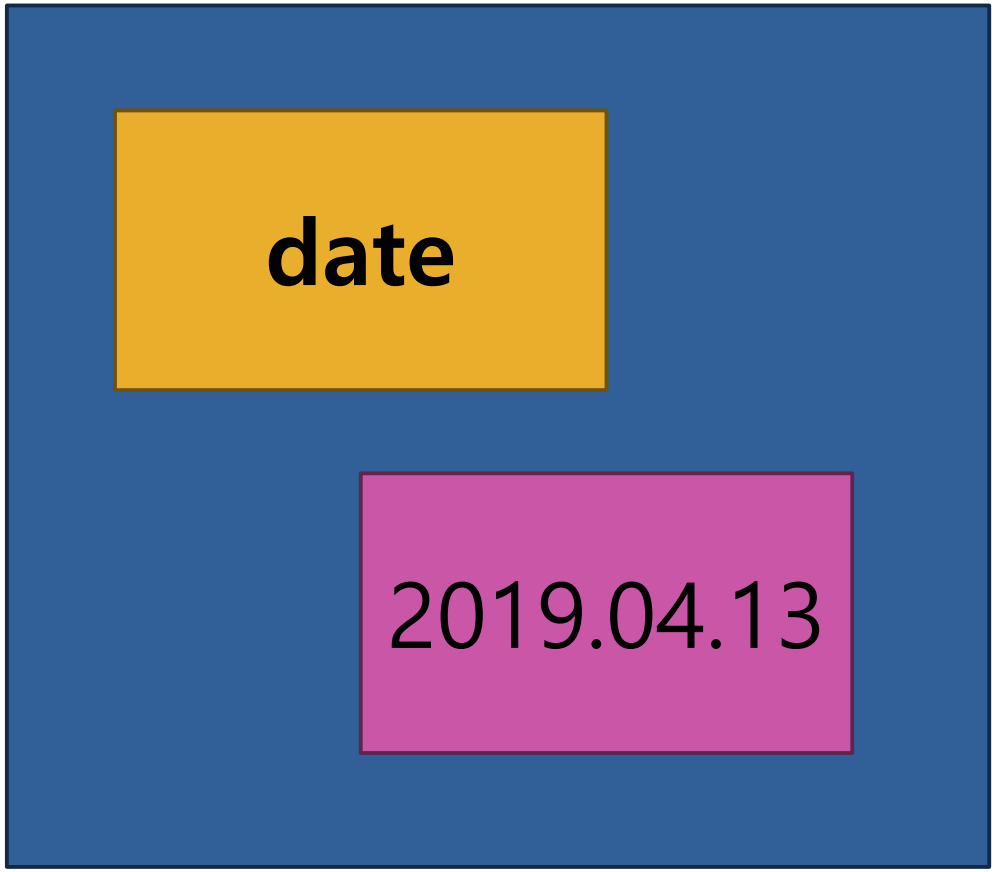


ItemType

date = 2019.04.13

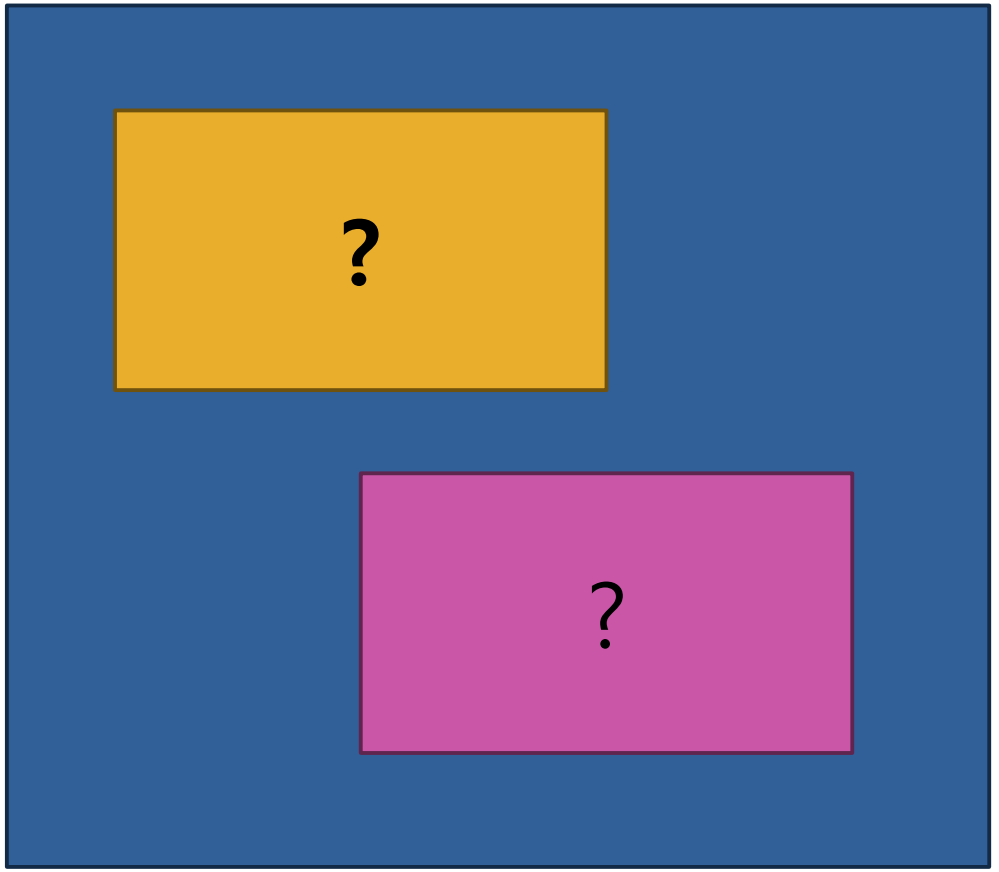
- ItemType<std::string> x ("date", "2019.04.13");

std::cout << x.ToString();



ItemType

"Hello Wiz"

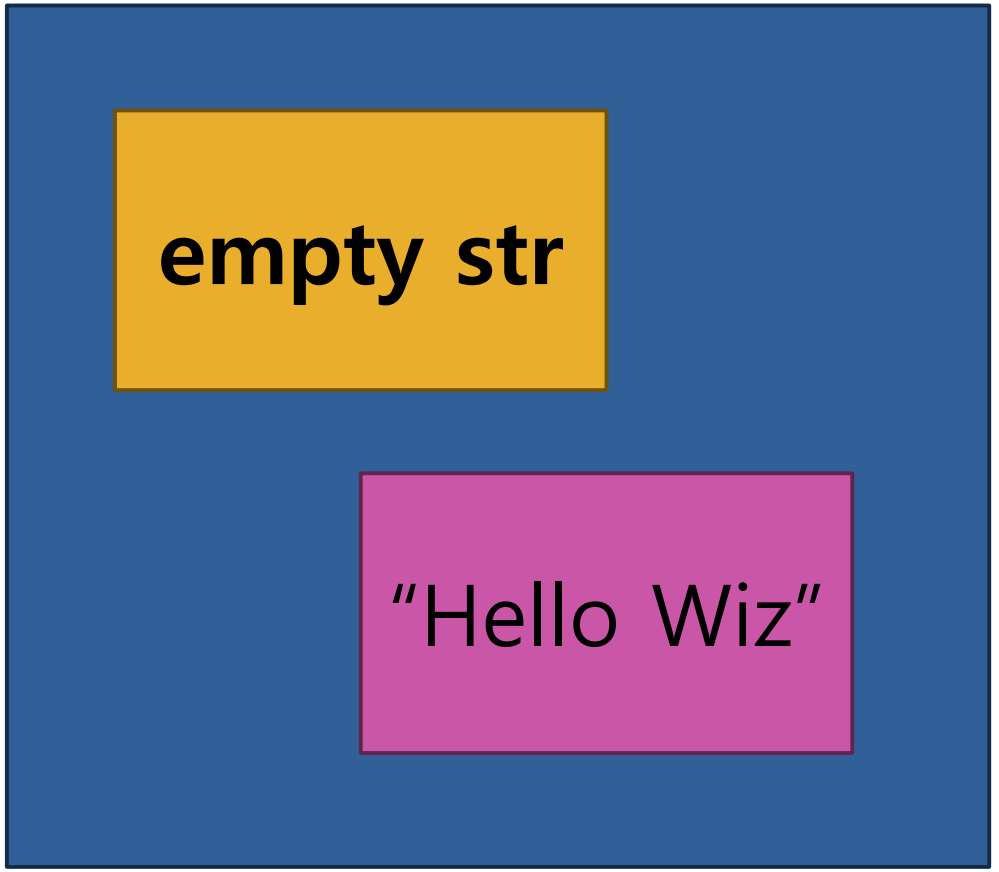


ItemType

"Hello Wiz"

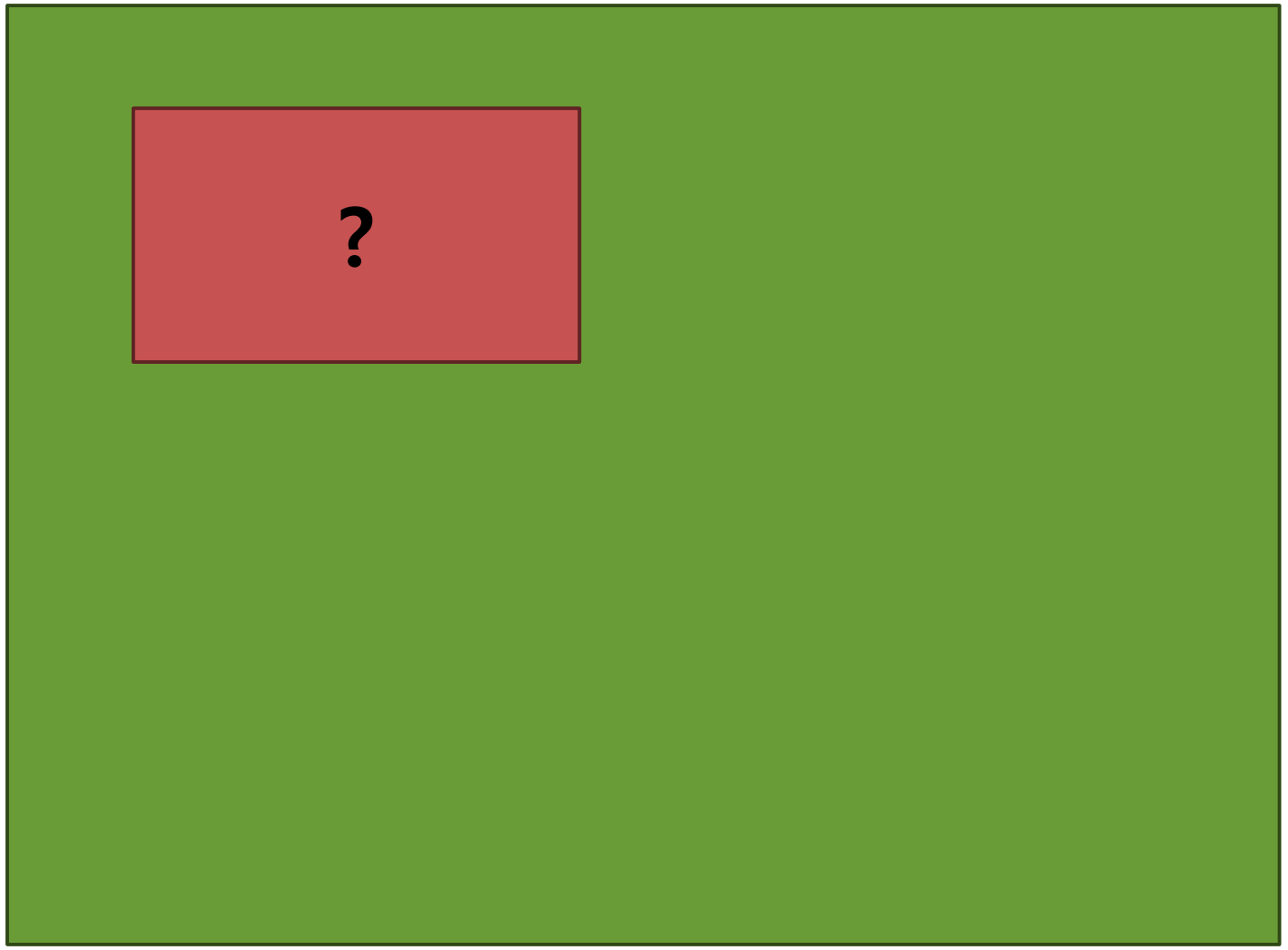
- ItemType<std::string> x ("", "Hello Wiz");

- std::cout << x.ToString();



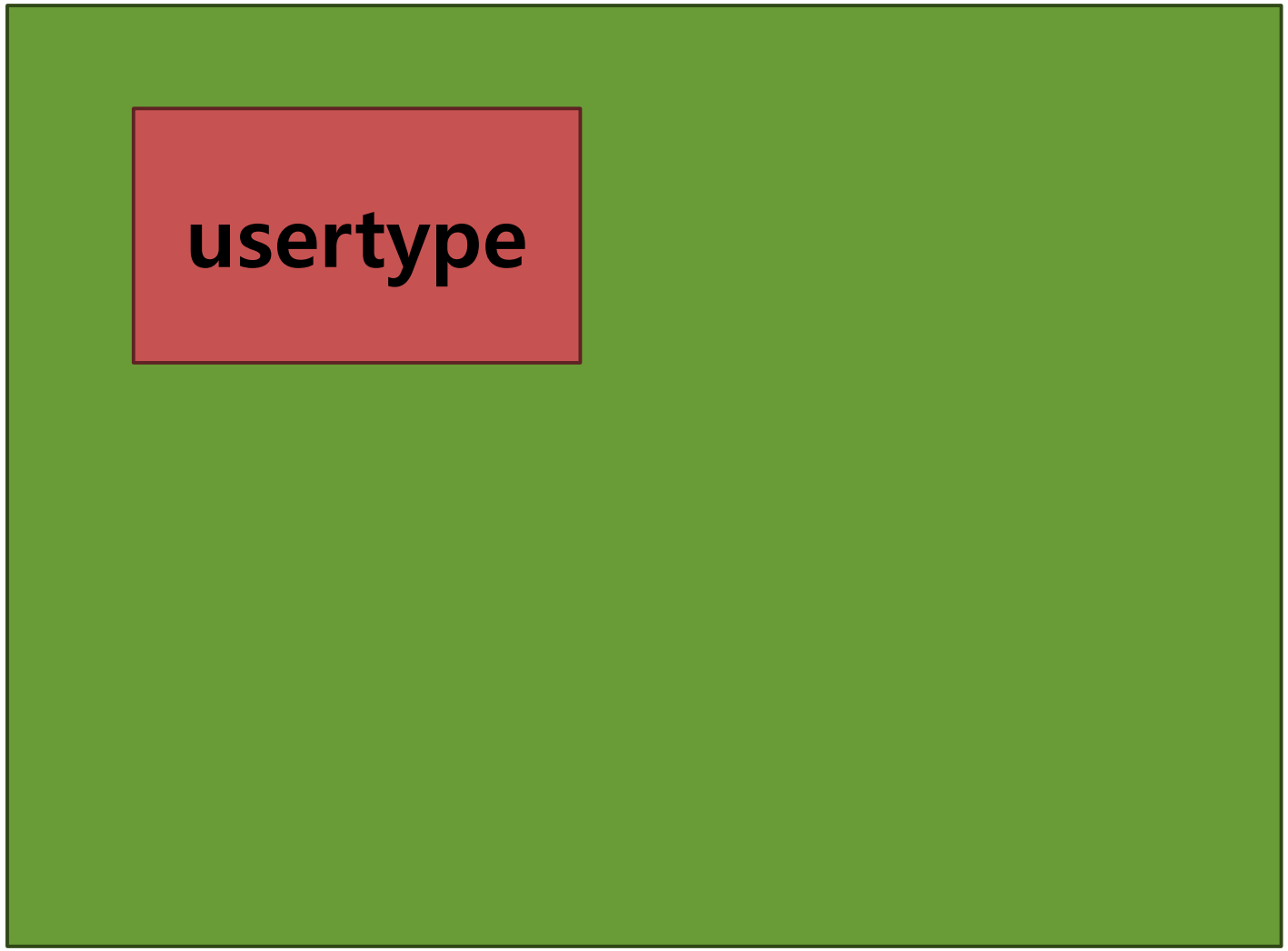
UserType

UserType



usertype = { }

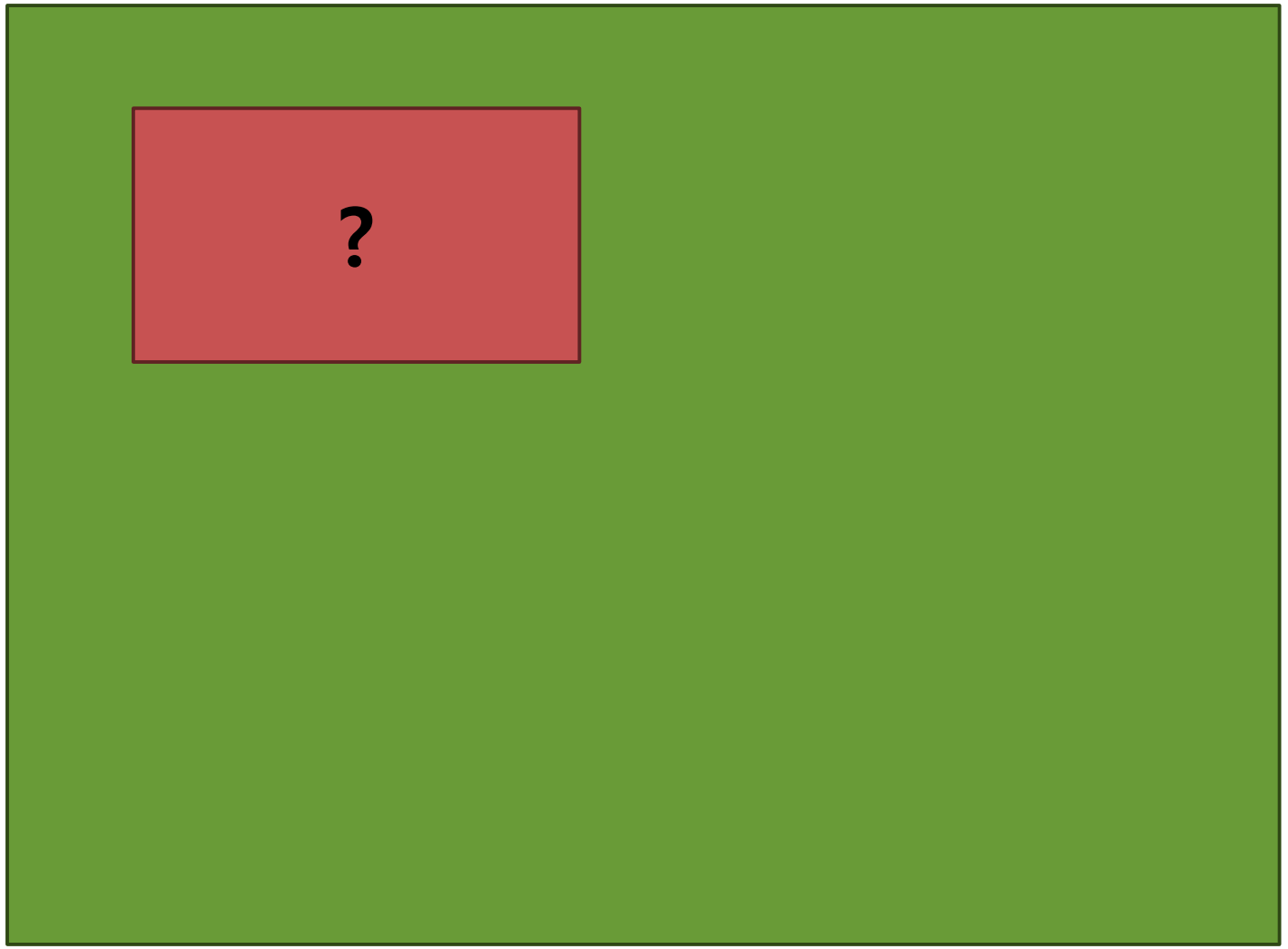
UserType



usertype = { }

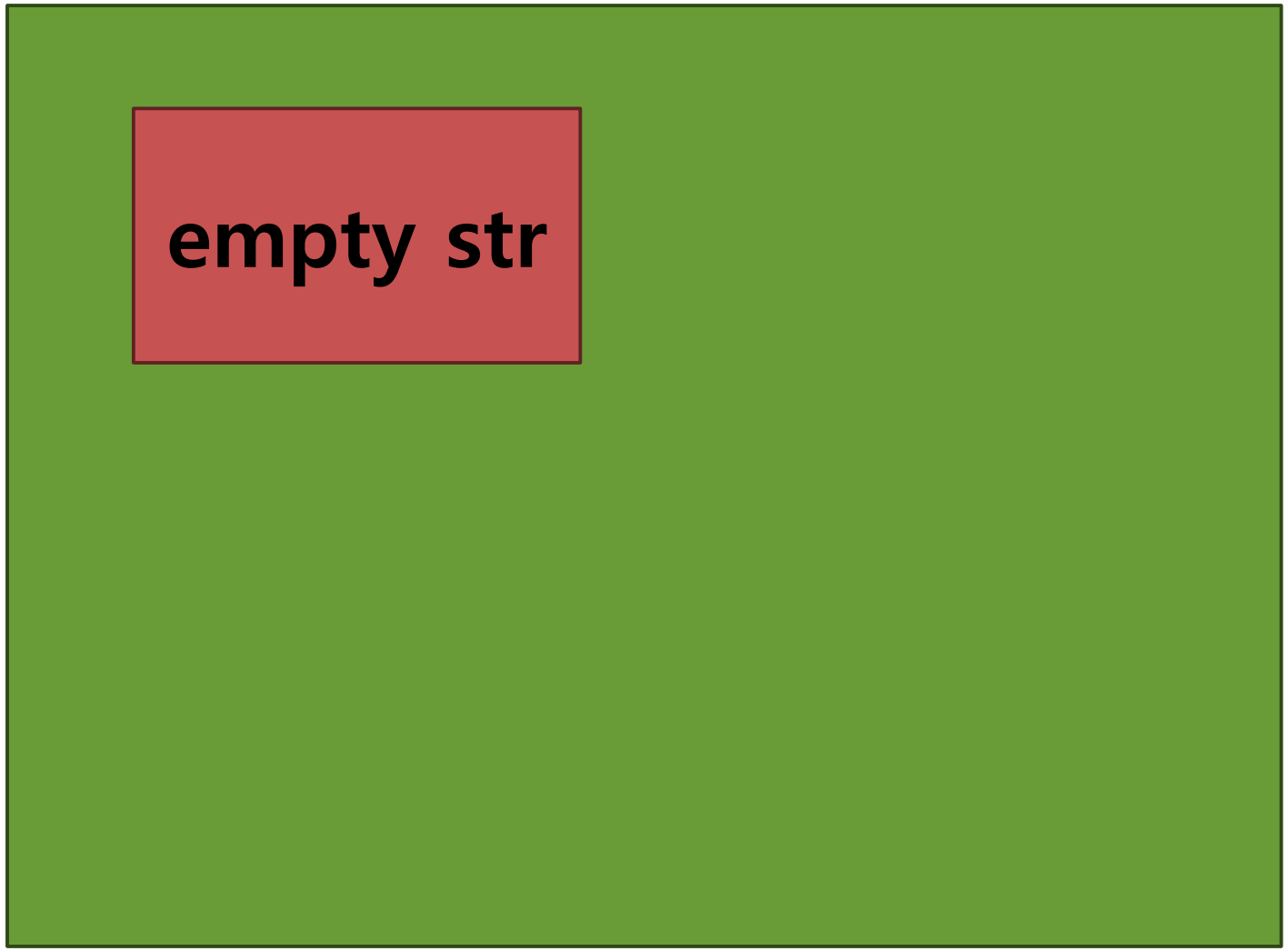
```
UserType global;
UserType ut("usertype");
global.AddUserTypeItem(ut);

std::cout << global.ToString();
```

UserType



{ }

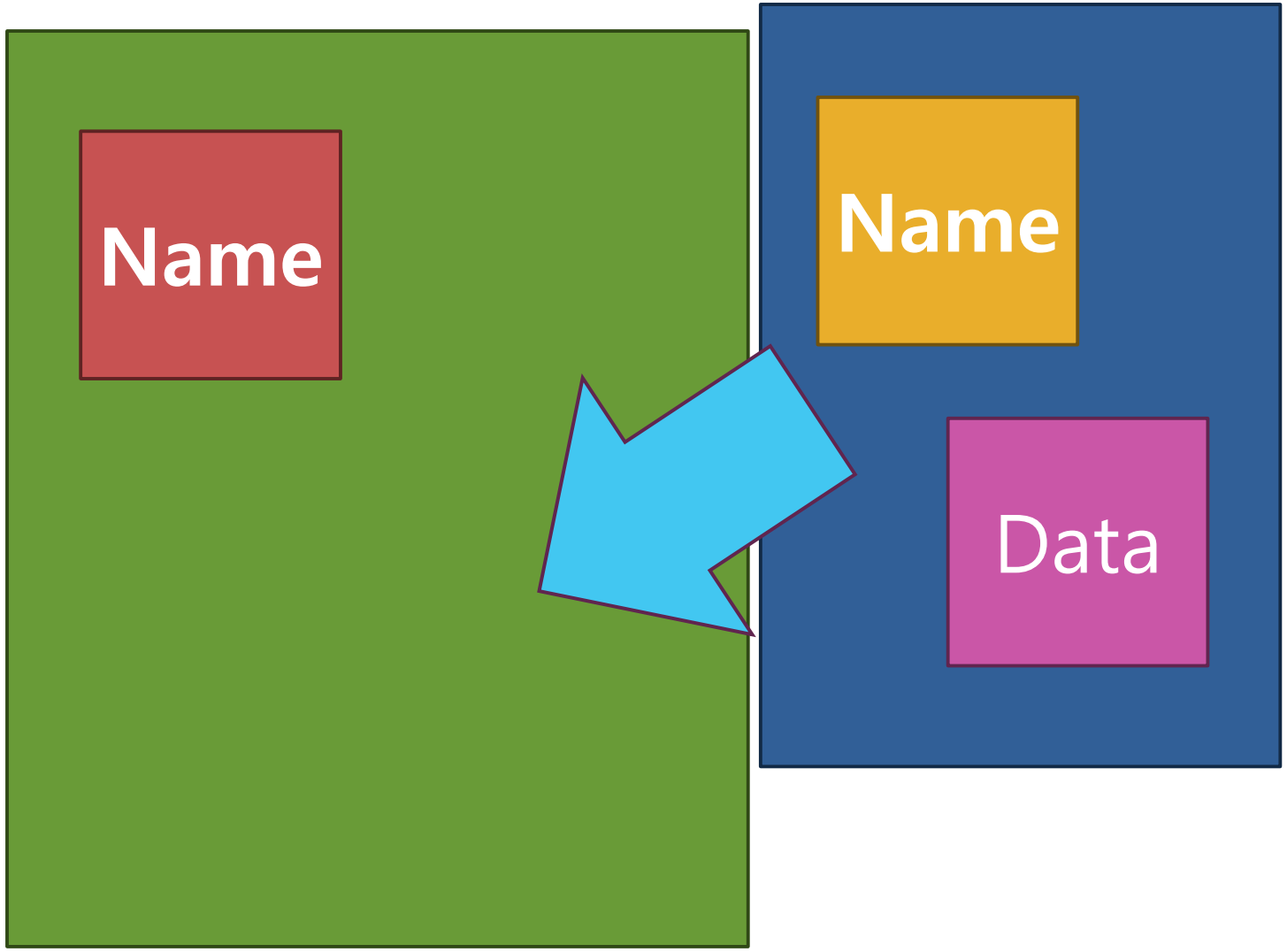
UserType



{ }

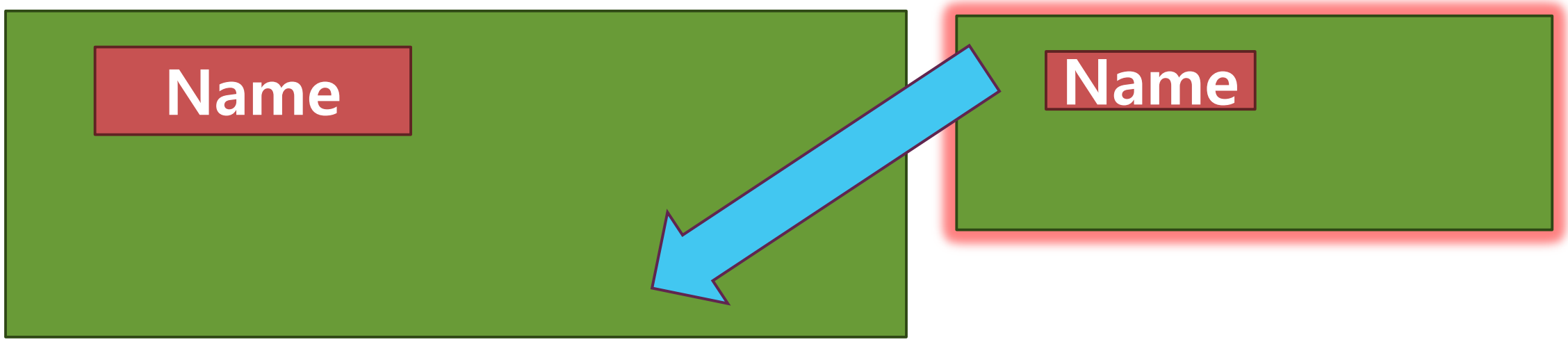
```
UserType global;
UserType ut("");
global.AddUserTypeItem(ut);

std::cout << global.ToString();
```

UserType can have ItemTypes




```
{
    wiz::UserType ut("Name");
    wiz::ItemType<std::string> it("Name", "Data");

    ut.AddItemType(it);
}
```

UserType can have (UserType*)s



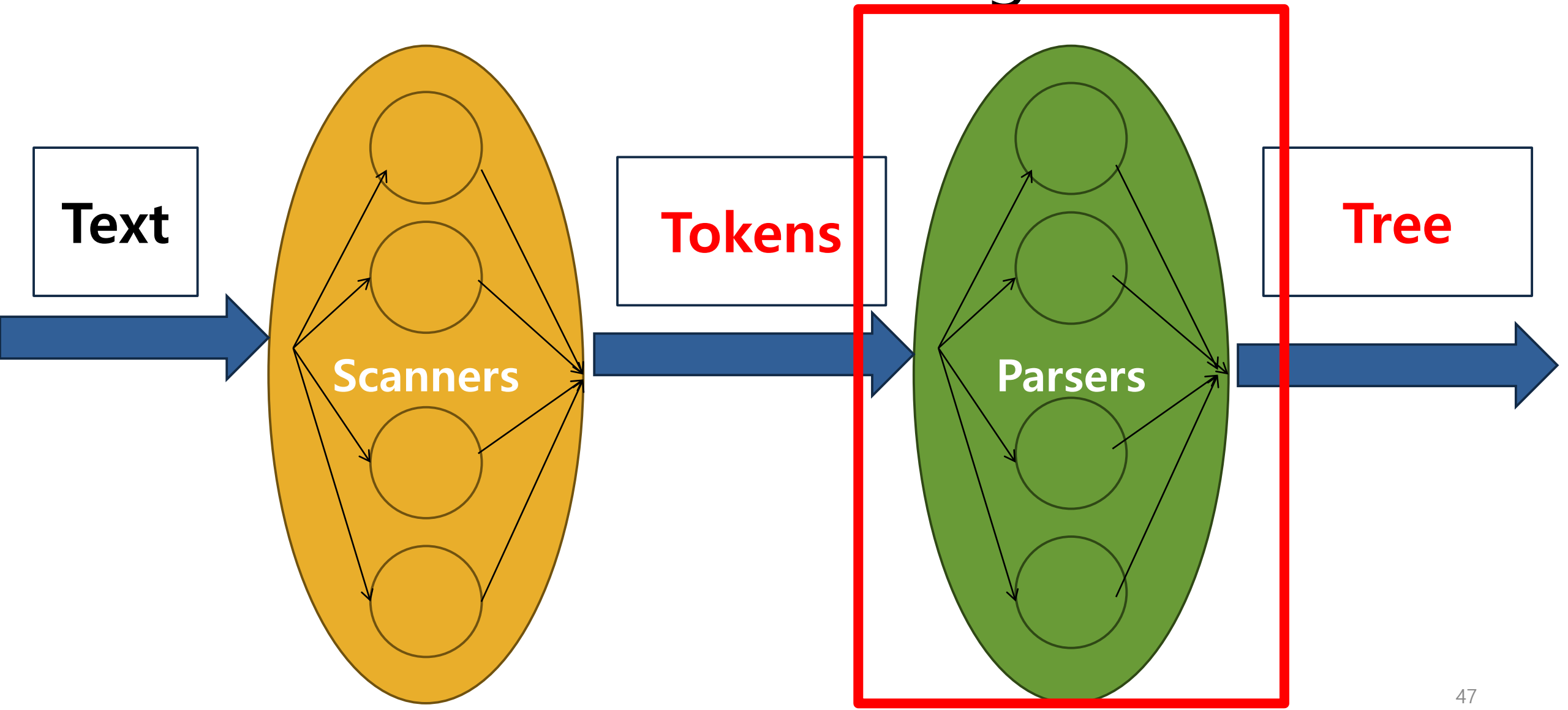

```
{
    wiz::UserType ut("Name");
    wiz::UserType global("Name");

    global.AddUserTypeItem(ut); // 내부에서 동적할당
}
```

Tree구조

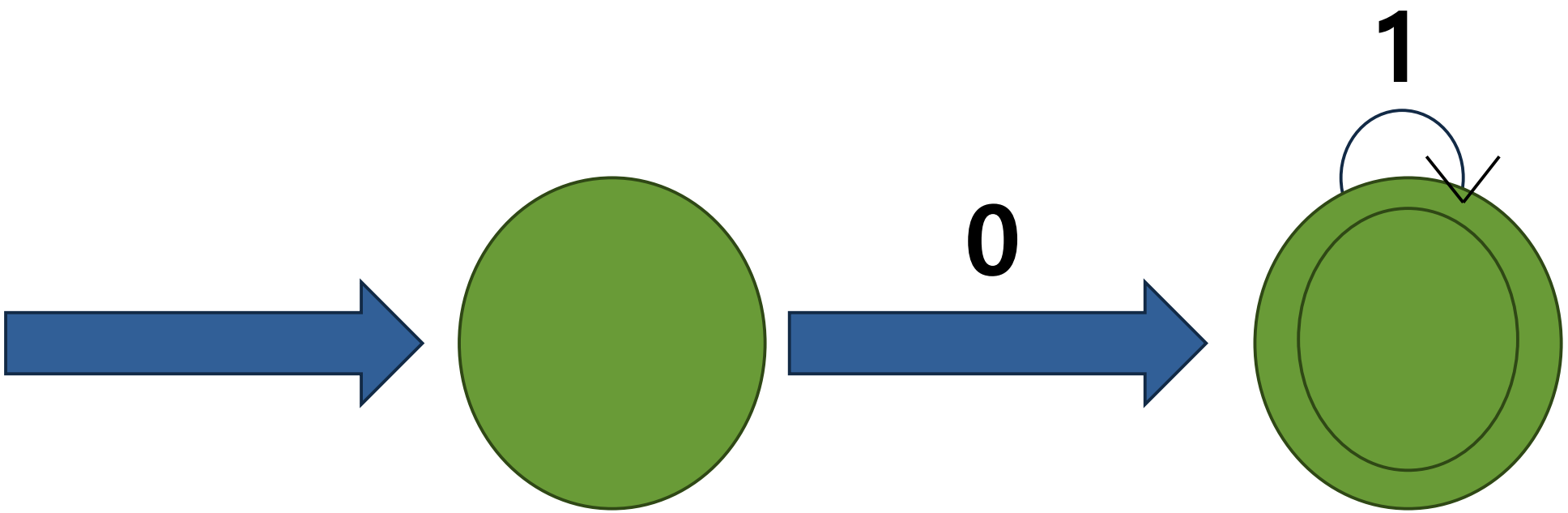
```
date=1947.5.8
save_game="Russia.eu4"
savegame_versions={
        "1.18.1.0"
        "1.18.2.0"
        "1.18.3.0"
        "1.18.4.0"
        "1.19.1.0"
}
```

```
{
    wiz::UserType global;
    global.AddItem("date", "1947.5.8");
    global.AddItem("save_game", "\"Russia.eu4\"");
    {
        wiz::UserType temp("savegame_versions");
        temp.AddItem("", "\"1.18.1.0\"");
        temp.AddItem("", "\"1.18.2.0\"");
        temp.AddItem("", "\"1.18.3.0\"");
        temp.AddItem("", "\"1.18.4.0\"");
        temp.AddItem("", "\"1.19.1.0\"");
        global.AddUserTypeItem(temp);
    }

    std::cout << global.ToString();
}
```

Parallel Parsing



Automata 0(1)*

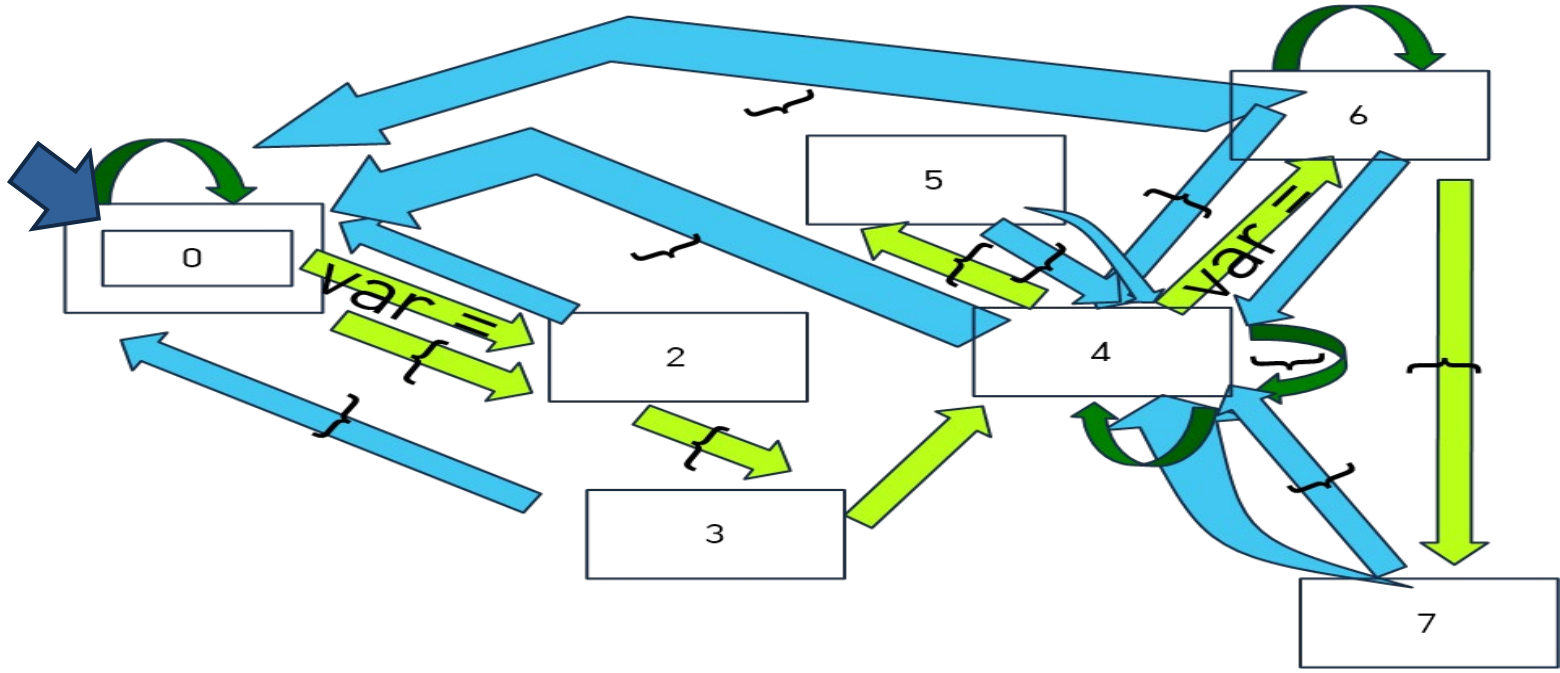


Parsing - Automata? (**Old** version)

} => state가 0 or 4

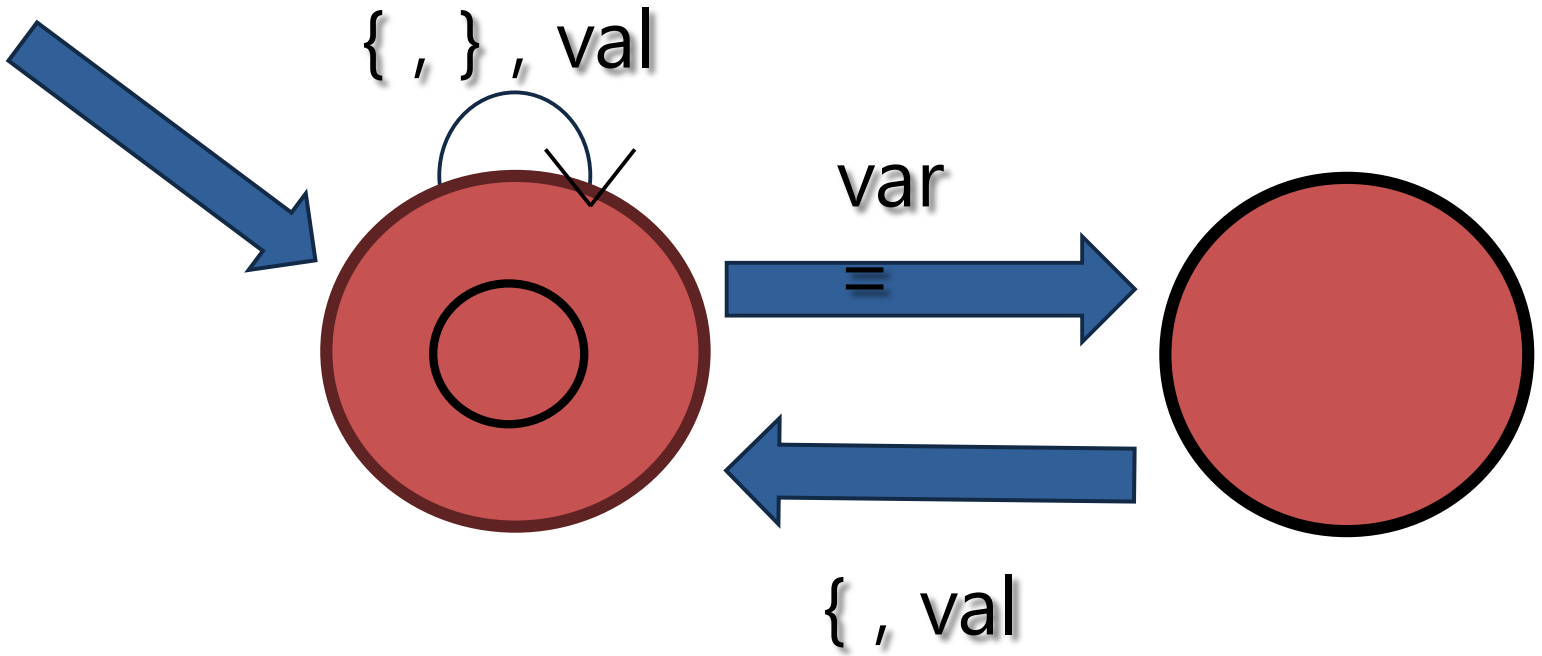
} 기준으로 토큰을 나누면 내부 state를 쉽게 알 수 있다.

- S -> A
- A -> var = val A |
  var = { A } A |
  { A } A |
  val A |
  λ
- var, val : 문자열!



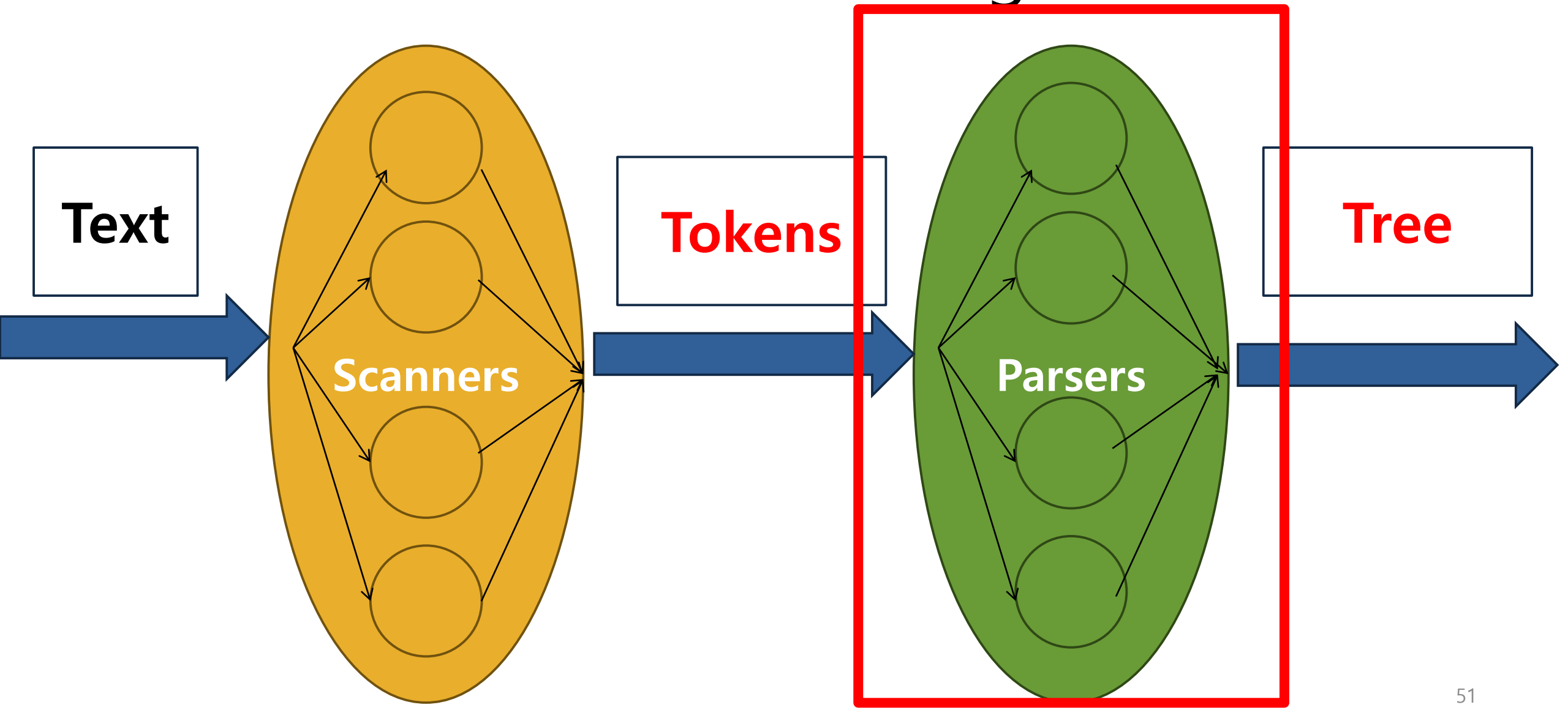
Parsing - Automata (**New**, Simple version!)

- S -> A
- A -> var = B |
  { A |
  } A |
  val A |
  λ
- B -> val A | { A



- var, val : 문자열!

Parallel Parsing



Parallel Parsing - Divide

- **depth**?

  initial depth = 0;

  if meet **{** : depth++;

  if meet **}** : depth--;

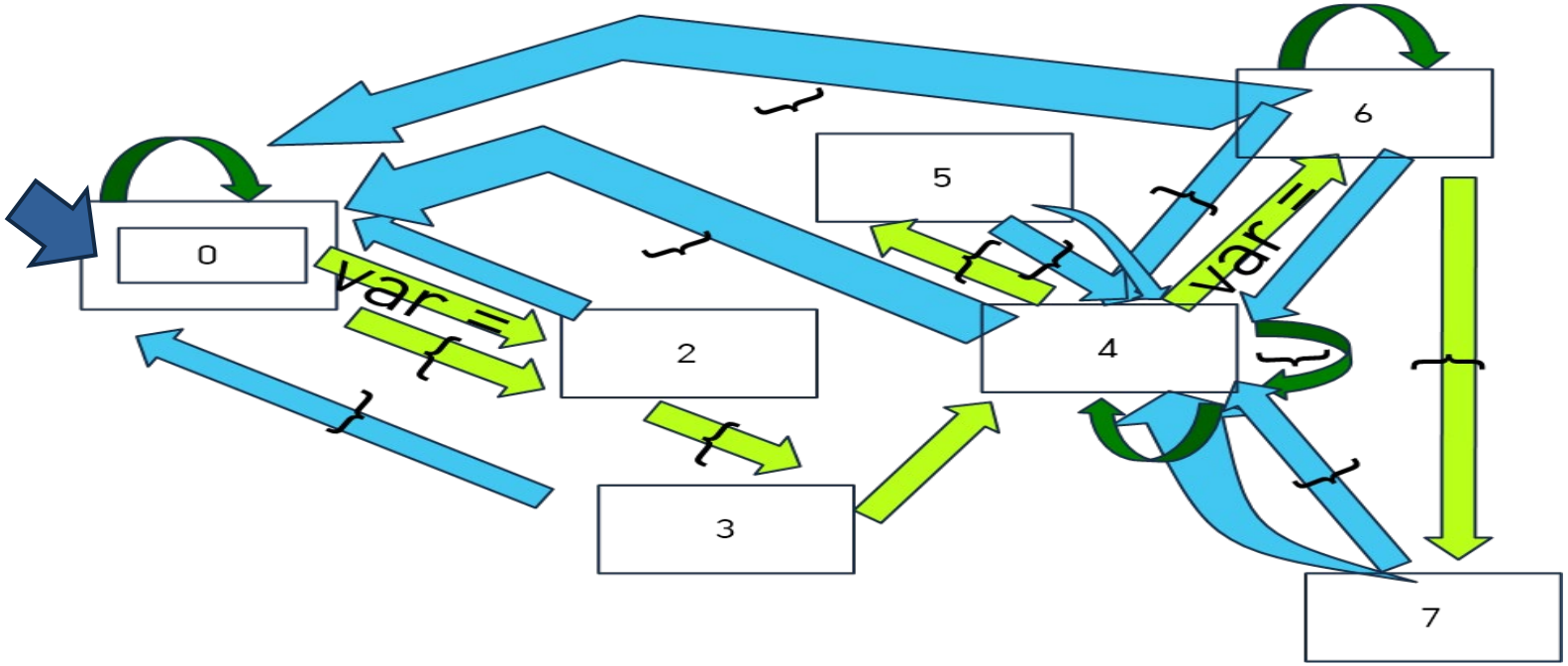
- A = { ~~~ } # here depth == 0

  B = { ~~~ } # here depth == 0

  C = { ~~~ } # here depth == 0 -> cf) 3 Files

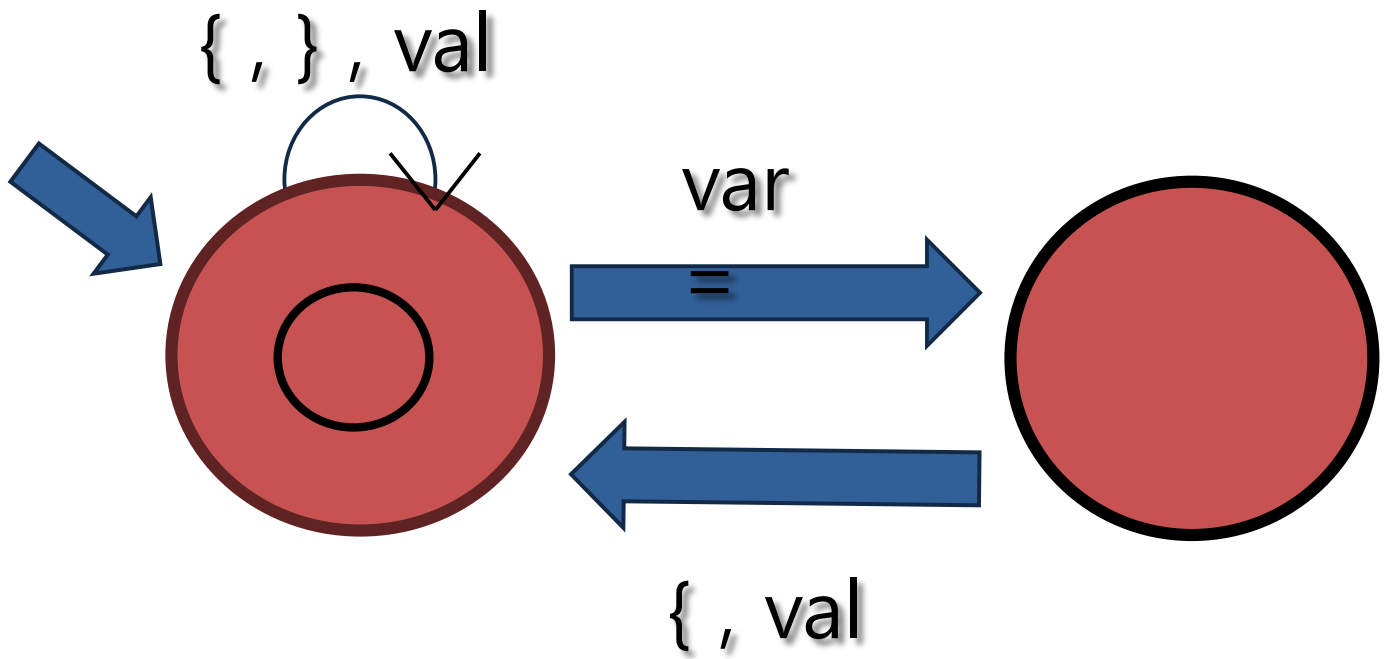
Parallel Parsing - Divide, **Old** Version

- A = {
  B = {
  1 2 3
  } # divided here?
  C = {
  4 5 6
  } # divided here?
  } # divided here?



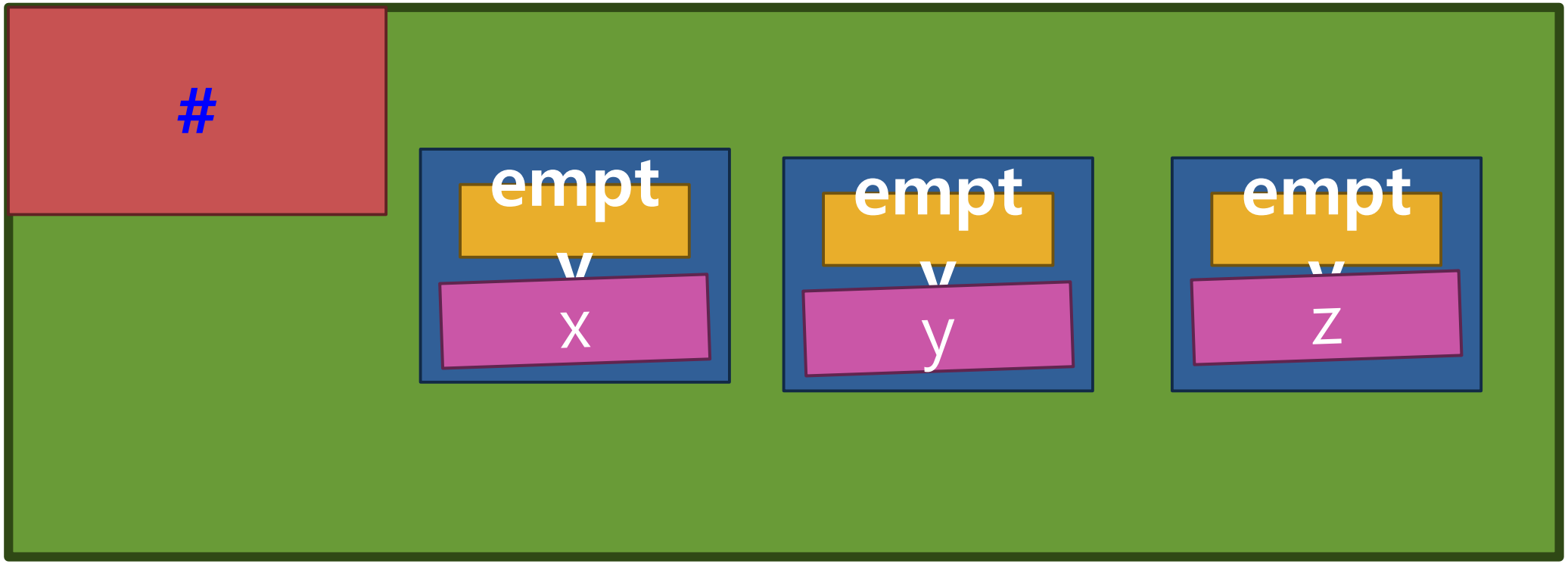
Parallel Parsing - Divide, **New** Version

```
1935.1.1={
        advisor={
                name="Fritjof Akeleye"
                type=naval_reformer
                skill=1
                location=1
                discount=no
                date=1935.1.1
                hire_date=1.1.1
                female=no
                id={
                        id=81577
                        type=51
                }
        }
}
```



Virtual Node

- x y z }
- => # = { x y z }



Virtual Node의 특성

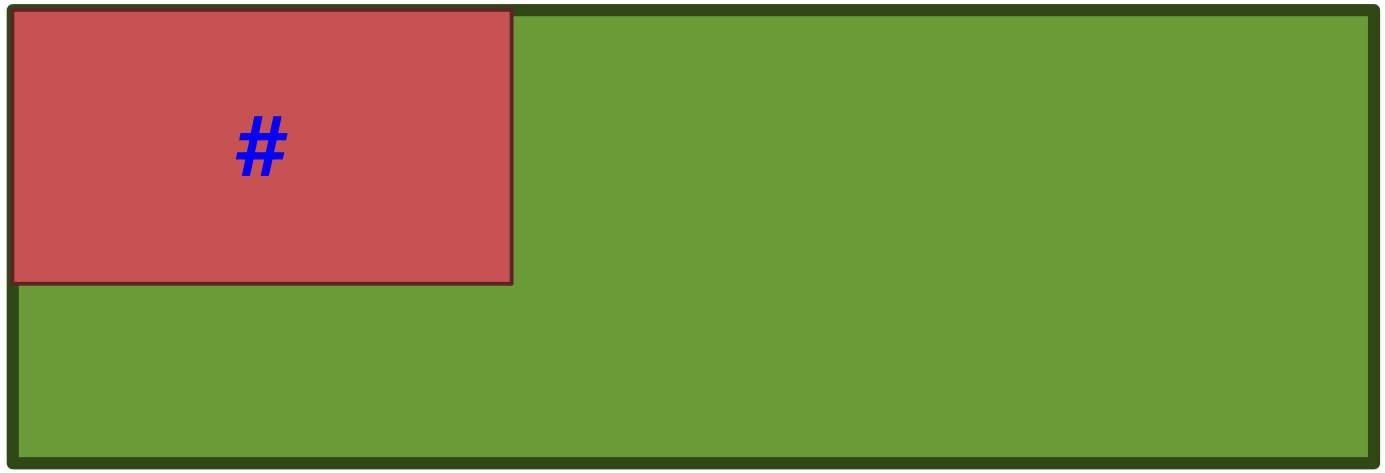
- # = {} # = {} : does not exist!
- # = { # = { # = { } } }

Virtual Node의 특성

- input }
- result # = { }

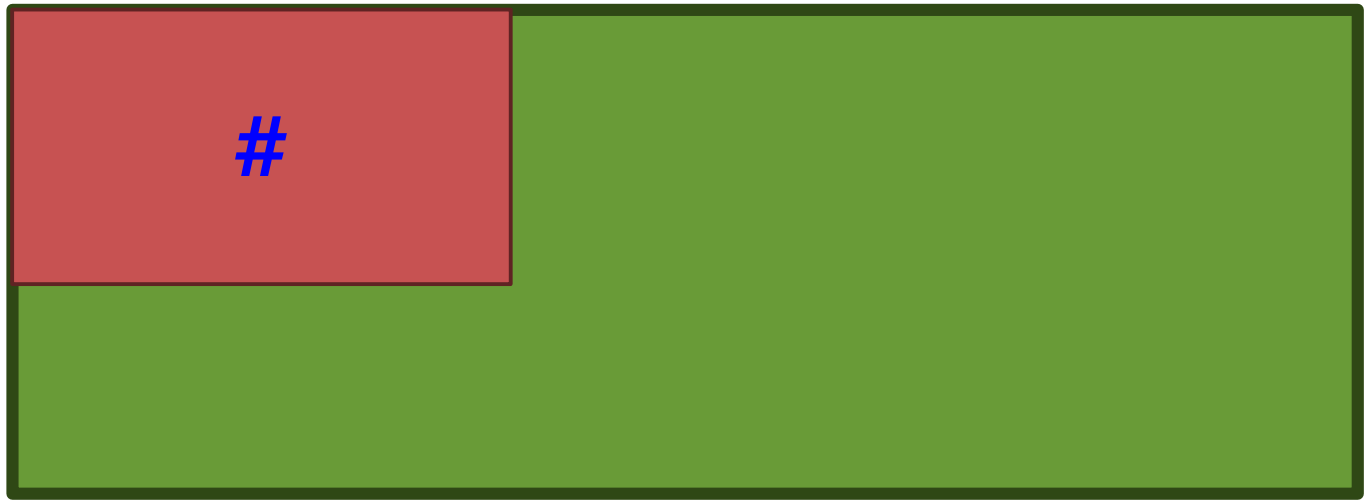
Virtual Node의 특성

- input } {
- result # = { } { }



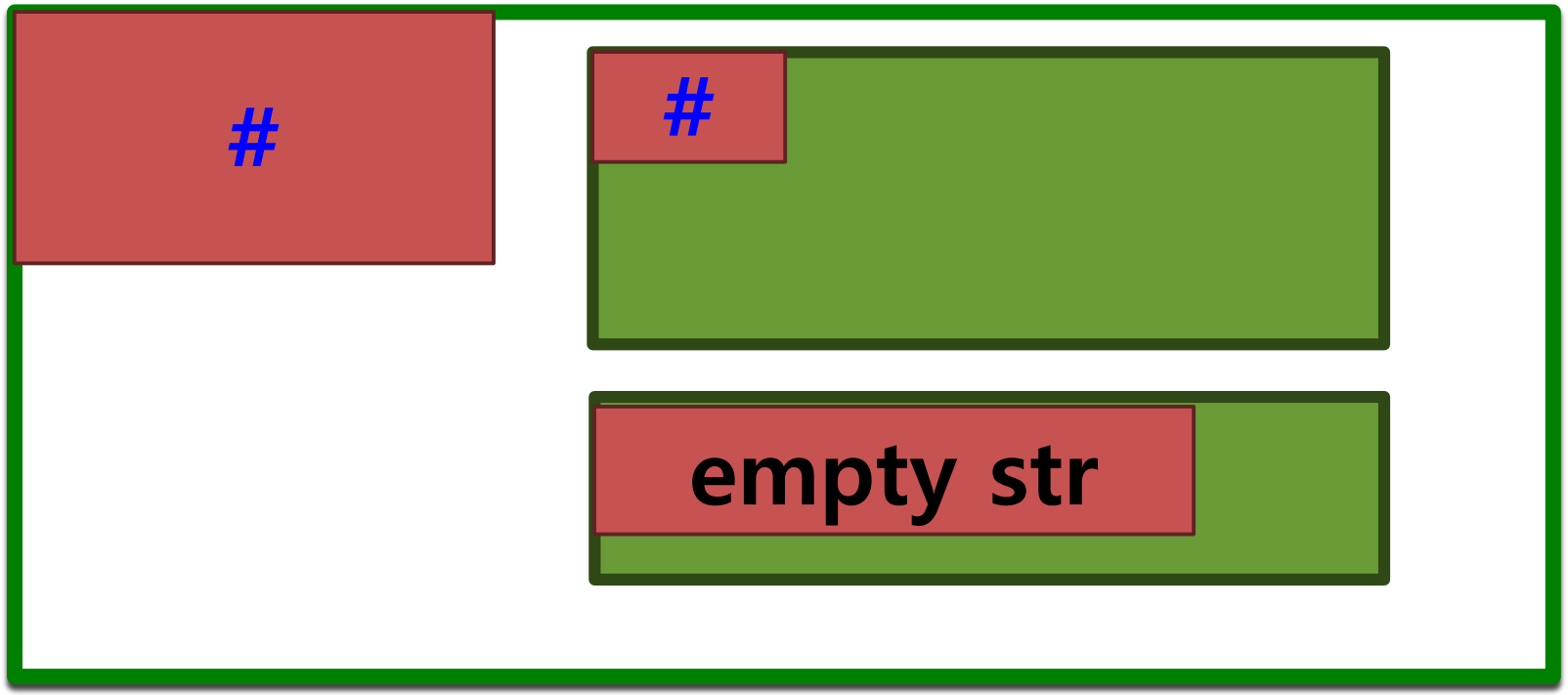
Virtual Node의 특성

- input } { }
- result # = { } { }

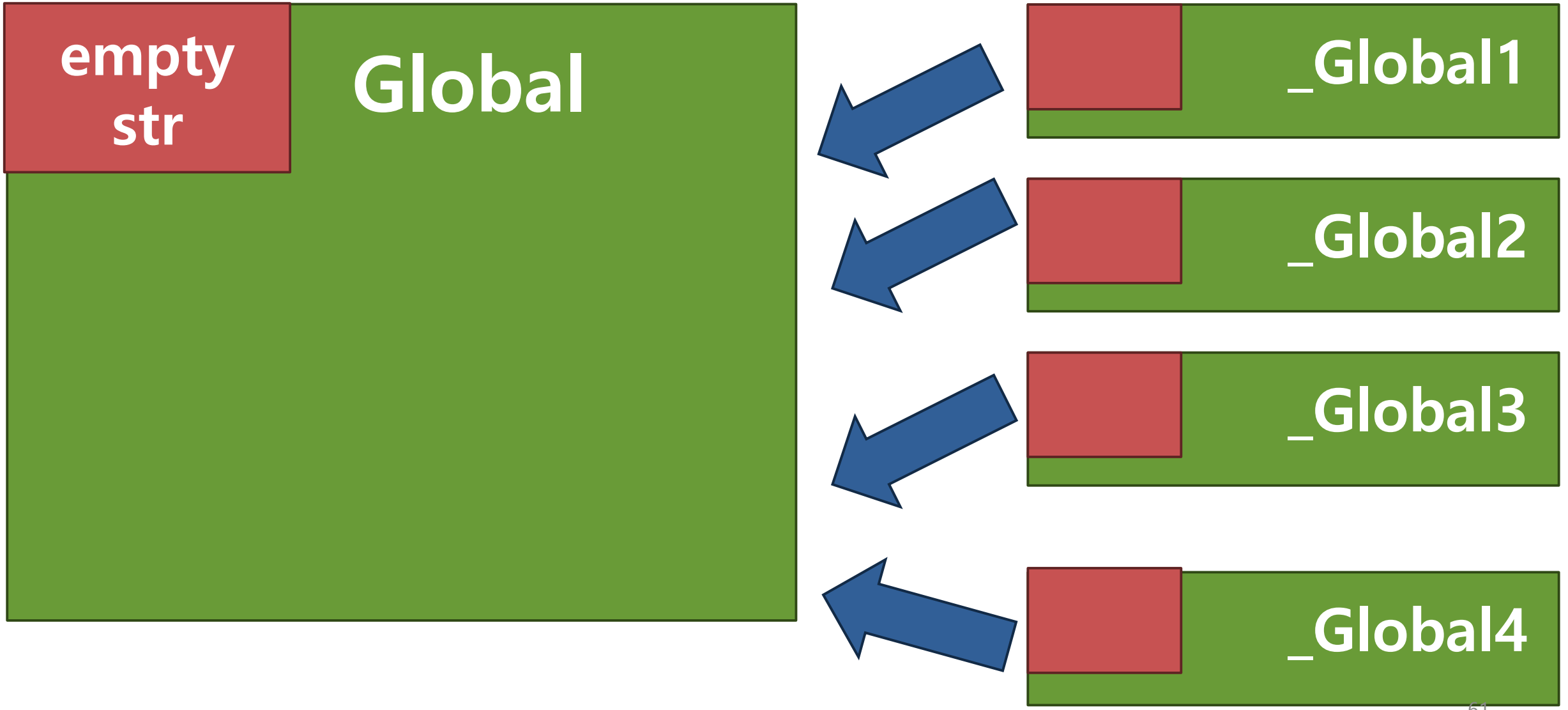


Virtual Node의 특성

- input } { } }
- result # = { # = { } { } }



Merge

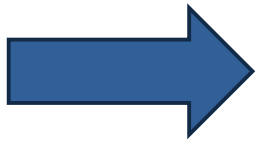


Merge

```
1935.1.1={
        advisor={
                name="Fritjof Akeleye"
                type=naval_reformer
        }
}
```

```
        skill=1
        location=1
        discount=no
```

```
                date=1935.1.1
                hire_date=1.1.1
                female=no
                id={
                }
```

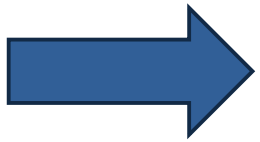


```
1935.1.1={
        advisor={
                name="Fritjof Akeleye"
                type=naval_reformer
                skill=1
                location=1
                discount=no
```

```
#={
        #={
                #= {
                        id=81577
                        type=51
                }
        }
}
```

Merge

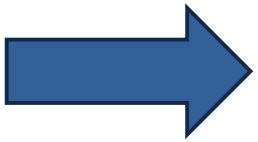
```
1935.1.1={
        advisor={
                name="Fritjof Akeleye"
                type=naval_reformer
                skill=1
                location=1
                discount=no


            date=1935.1.1
            hire_date=1.1.1
            female=no
            id={
            }

#={
        #={
                #= {
                        id=81577
                        type=51
                }
        }
}
```



```
1935.1.1={
        advisor={
                name="Fritjof Akeleye"
                type=naval_reformer
                skill=1
                location=1
                discount=no
                date=1935.1.1
                hire_date=1.1.1
                female=no
                id={
```

Merge

```
1935.1.1={
        advisor={
                name="Fritjof Akeleye"
                type=naval_reformer
                skill=1
                location=1
                discount=no
                date=1935.1.1
                hire_date=1.1.1
                female=no
                id={
                }
        #={
                #= {
                        id=81577
                        type=51
                }
        }
```



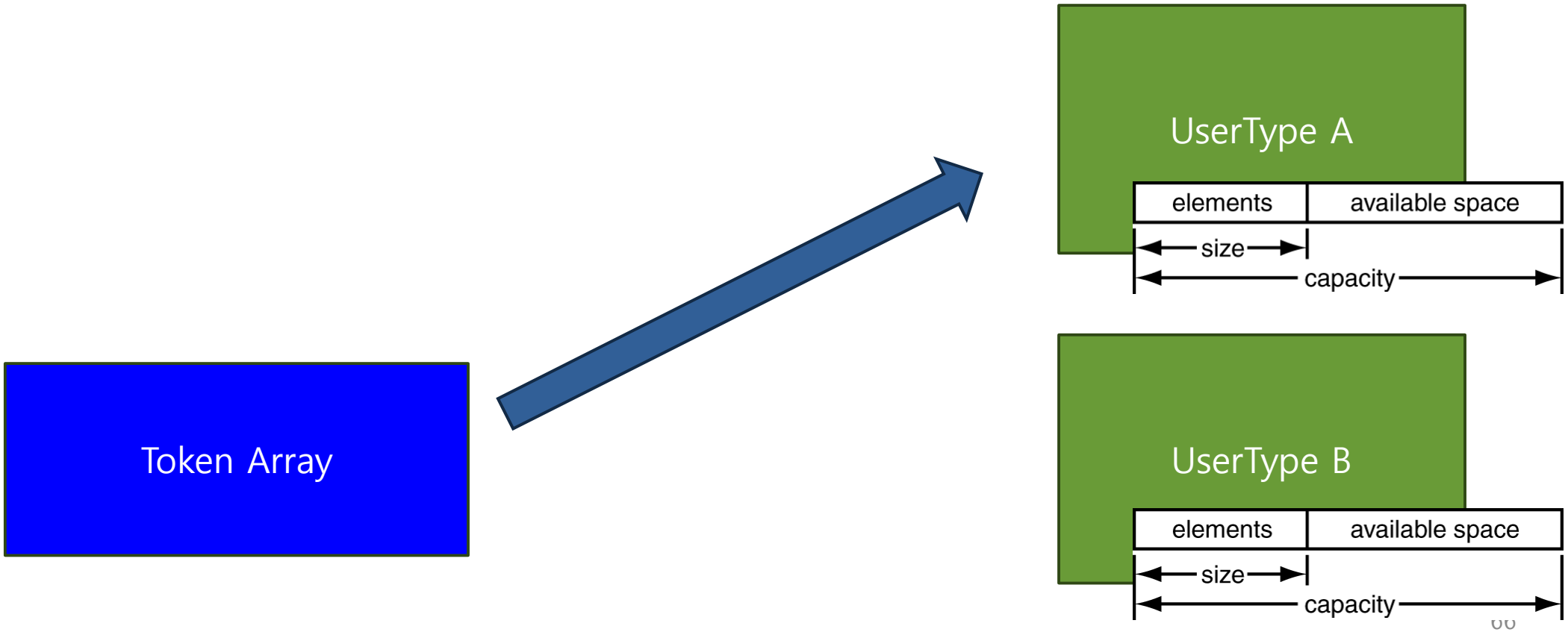
```
1935.1.1={
        advisor={
                name="Fritjof Akeleye"
                type=naval_reformer
                skill=1
                location=1
                discount=no
                date=1935.1.1
                hire_date=1.1.1
                female=no
                id={
                        id=81577
                        type=51
                }
        }
}
```

reserve를 활용하자 + push_back

```
        ledger_data_y=
        {
0 0 0 0 0 1 1 1 1 2 2 2 3 3 4 4 4 4 4 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 8 8 8 8 8 8 8 9 9 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0          }
        ledger_data_x=
        {
1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418
1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437
1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456
1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475
1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494
1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513
1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532
1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551
1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570
1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589
1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608
1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627
1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646
1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665
1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684
1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703
1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722
1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741
1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760
1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779
1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798
1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817
1818 1819          }
        name="KHD"
```

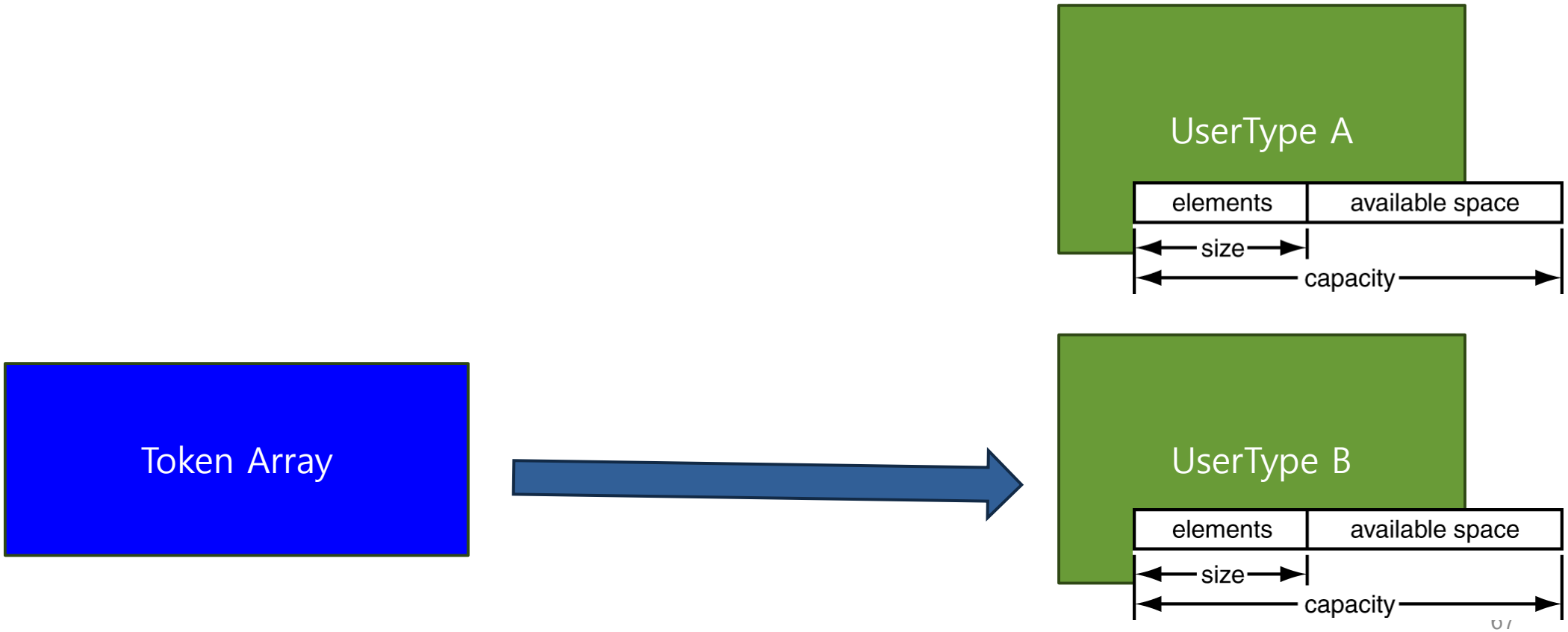
reserve를 활용하자 + push_back

- A = { 1 2 3 … 1024 } B = { 1 2 3 … 1024 }



reserve를 활용하자 + push_back

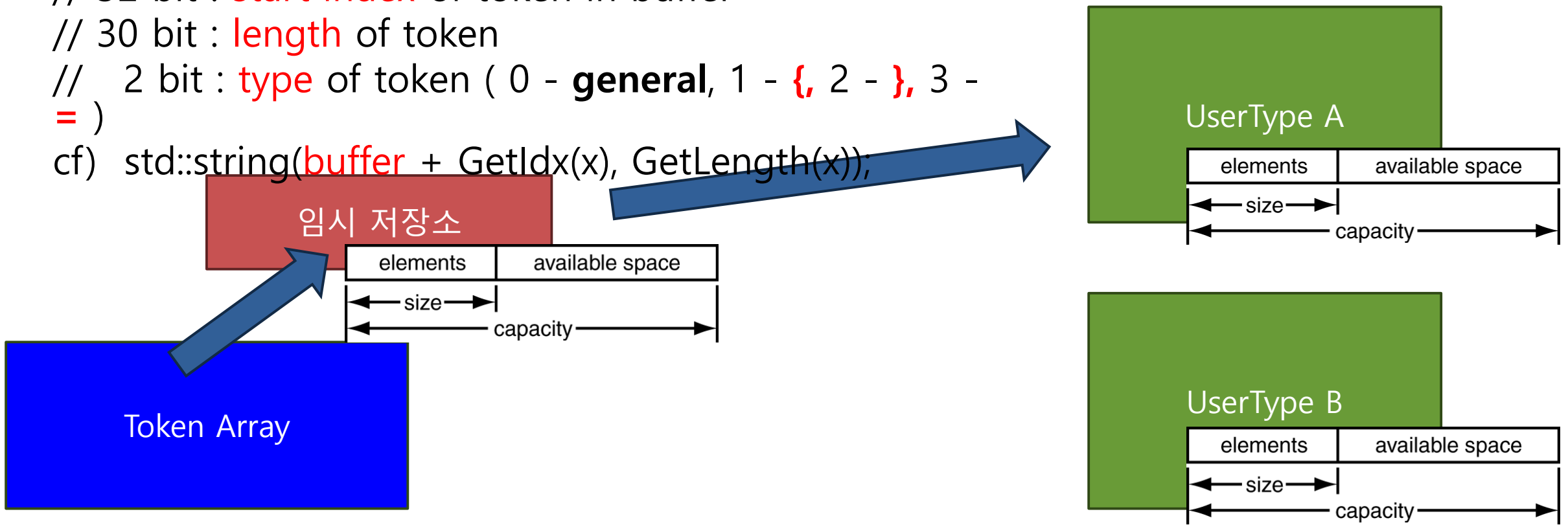
- A = { 1 2 3 ... 1024 } B = { 1 2 3 ... 1024 }



reserve를 활용하자 + push_back

- A = { 1 2 3 ... 1024 } B = { 1 2 3 ... 1024 }

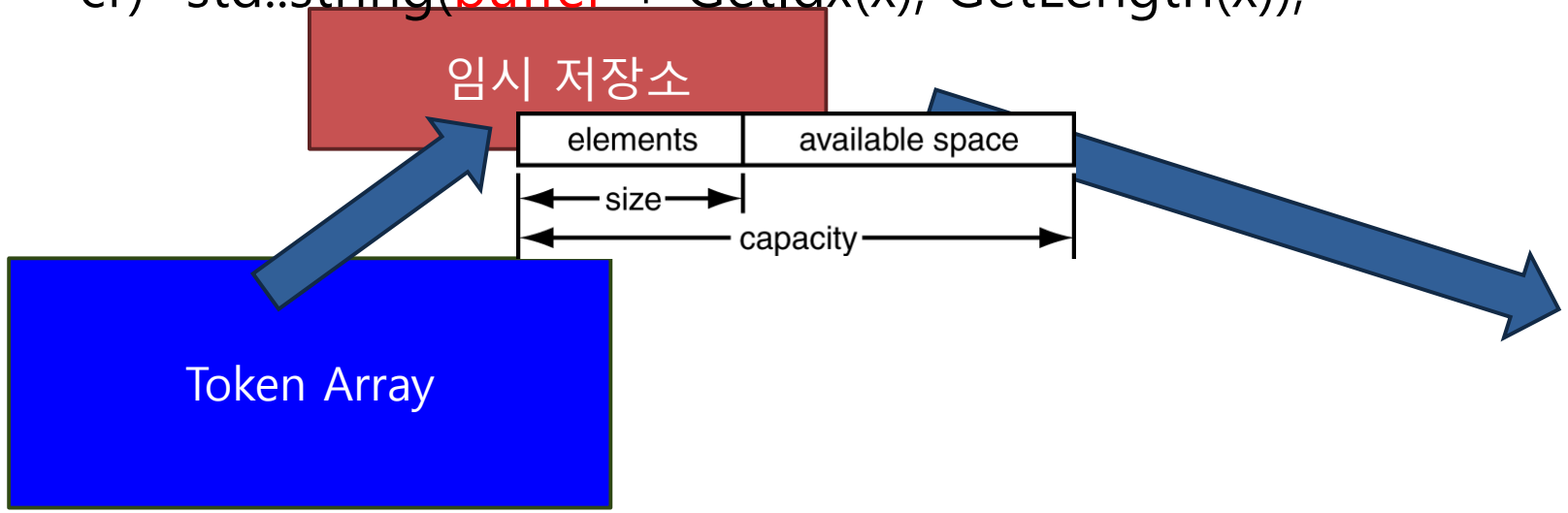
```
long long x;
// 32 bit : start index of token in buffer
// 30 bit : length of token
//   2 bit : type of token ( 0 - general, 1 - {, 2 - }, 3 - = )
cf)  std::string(buffer + GetIdx(x), GetLength(x));
```

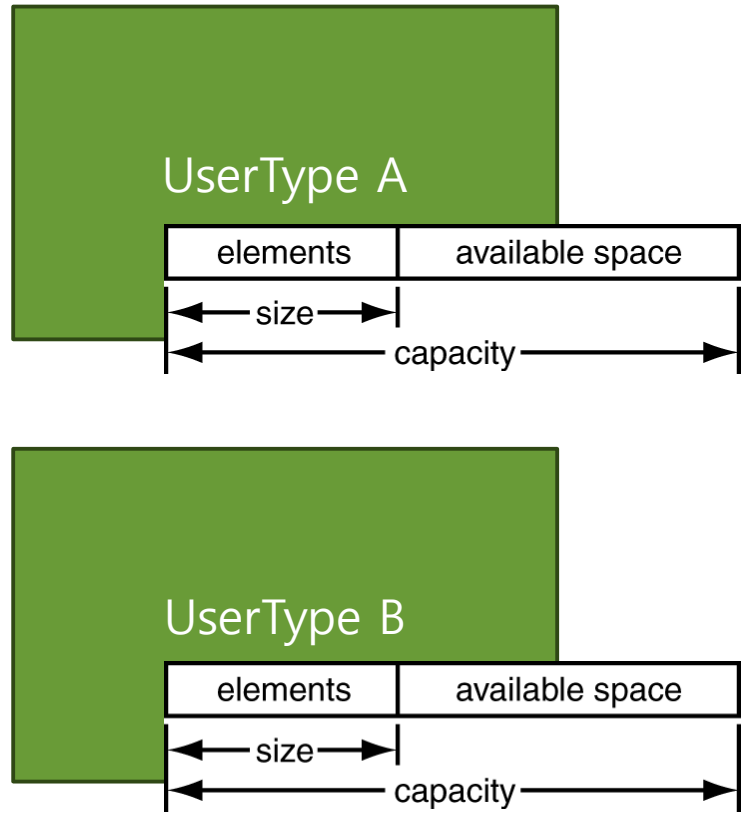


reserve를 활용하자 + push_back

- A = { 1 2 3 ... 1024 } B = { 1 2 3 ... 1024 }

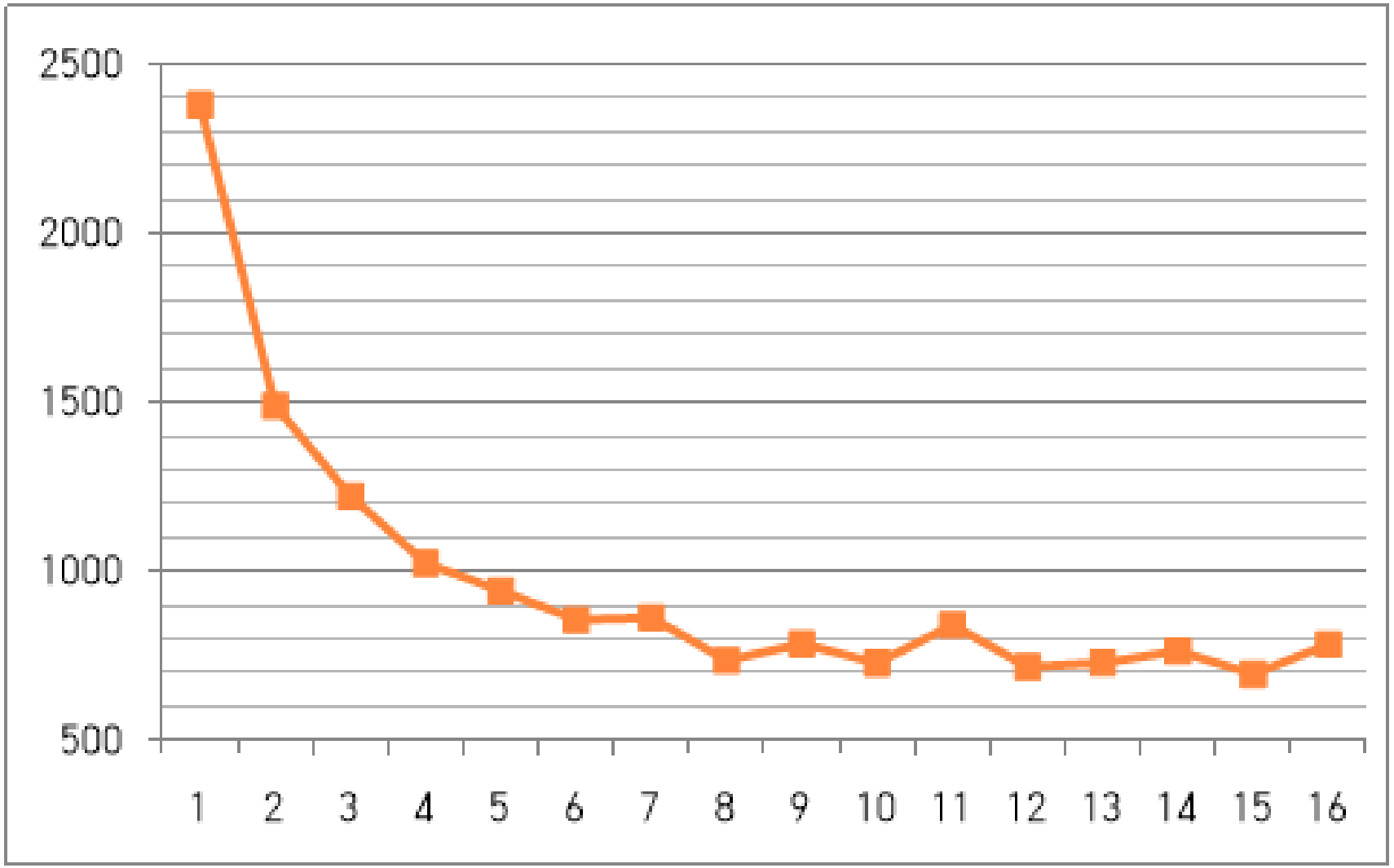
```
long long x;
// 32 bit : start index of token in buffer
// 30 bit : length of token
//   2 bit : type of token ( 0 - general, 1 - {, 2 - }, 3 - = )
cf)  std::string(buffer + GetIdx(x), GetLength(x));
```





Thread개수별 걸린시간(ms) 비교

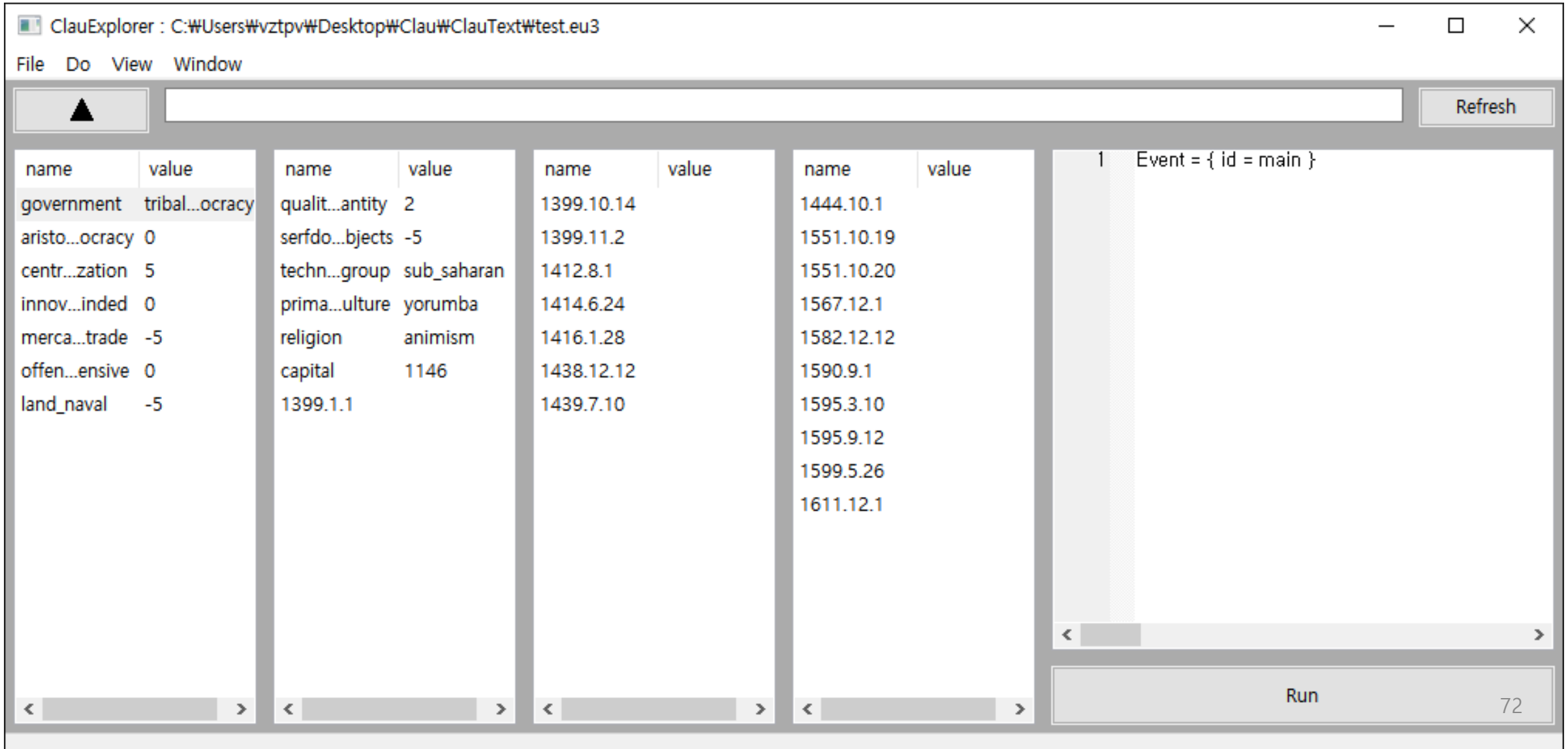
Intel 제온 프로세서
가상 CPU코어 24개
윈도우즈 서버 OS
SSD
Parsing 50MB Data



Parallel Paradox Parser의 문제점

- 다른 프로세스가 CPU를 차지하는 정도에 따라 성능차이가 크다.
- 예외처리?
- UTF-8?

Related Project - ClauExplorer



Related Project - ClauText

```
Main = {
    $call = { id = start }
}

Event = {
    id = start

    $print = { { "Hello ClauText World" } }
    $print = { { \n } }

    $_getch = { } # pause
}
```

Related Project - ClauText

```
$if = {
    condition = {
        $AND_ALL = { $EQ = { TRUE $parameter.is_user_type }
            $COMP> = { $remove_quoted = { $parameter.name } 1444.1.1 }
            $is_quoted_str = { $parameter.name }
        }
    }
    then = {
        $assign = { $local.name data = { "1444.1.1" } }

        $return = { name = $local.name }
    }
}
```

Related Project - ClauSemanticValidator in Clautext

```
schema = {
    $Event = {
        id = test
        $parameter = { x y select input }
        $if = {
            cond = { $COMP< = { $parameter.x $parameter.input } }
            then = { $return = { TRUE } }
        }
        $return = { FALSE }
    }
    %order_on
    a = %int%event_test@'x=121 y=3'
    b = %int
}
data = {
    a = 120
    b = 124
}
```

Usage - Load Data From File

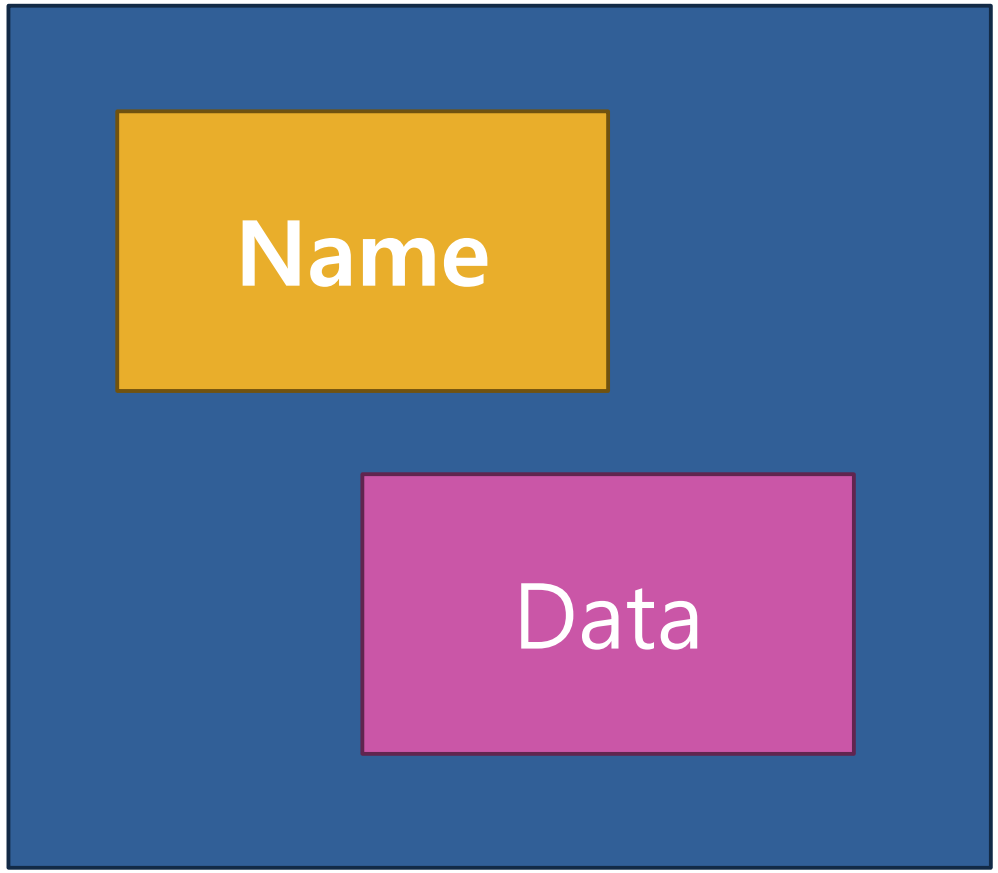
```
std::string fileName;

wiz::UserType global;

std::cin >> fileName;

int a = clock();
wiz::LoadData::LoadDataFromFile(fileName, global, 0, 0);
int b = clock();

std::cout << b - a << "ms" << "\n";
```

Usage - EU3 To Vic2

```
void AddCategoryToRegimentTypeMap(wiz::load_data::UserType* obj, RegimentCategory category, std::string categoryName, RegimentTypeMap& rtm)
{
    std::vector<wiz::load_data::UserType*> top = obj->GetUserTypeItem(categoryName);
    if (top.size() != 1)
    {
        LOG(LogLevel::Error) << "Could not get regiment type map for " << categoryName;
        exit(1);
    }
    std::vector<wiz::load_data::ItemType<wiz::DataType>> types = top[0]->GetItem("");
    if (types.size() == 0)
    {
        LOG(LogLevel::Error) << "No regiment types to map for " << categoryName;
        exit(1);
    }
    for (std::vector<wiz::load_data::ItemType<wiz::DataType>>::iterator itr = types.begin(); itr != types.end(); ++itr)
    {
        std::string type = (*itr).GetName().ToString();
        std::string strength = (*itr).Get(0).ToString();
        rtm[type] = std::pair<RegimentCategory, int>(category, atoi(strength.c_str()));
    }
}
```

Usage - EU3 To Vic2

```
int unitStrength = 0;
std::vector<wiz::load_data::ItemType<wiz::DataType>> strObj;
strObj = obj.GetItem("maneuver");
if (strObj.size() > 0)
    unitStrength += strObj[0].Get(0).ToInt();
strObj = obj.GetItem("offensive_morale");
if (strObj.size() > 0)
    unitStrength += strObj[0].Get(0).ToInt();
strObj = obj.GetItem("defensive_morale");
if (strObj.size() > 0)
    unitStrength += strObj[0].Get(0).ToInt();
strObj = obj.GetItem("offensive_fire");
if (strObj.size() > 0)
    unitStrength += strObj[0].Get(0).ToInt();
strObj = obj.GetItem("defensive_fire");
if (strObj.size() > 0)
    unitStrength += strObj[0].Get(0).ToInt();
strObj = obj.GetItem("offensive_shock");
if (strObj.size() > 0)
    unitStrength += strObj[0].Get(0).ToInt();
strObj = obj.GetItem("defensive_shock");
if (strObj.size() > 0)
    unitStrength += strObj[0].Get(0).ToInt();
strObj = obj.GetItem("hull_size");
if (strObj.size() > 0)
    unitStrength += strObj[0].Get(0).ToInt();
```



Etc..

- int to
  - star
- binary
  - use
  - strc

```
    int left = 0, right = arr.size() - 1;
    int middle = (left + right) / 2;

    while (left <= right) {
        const int dif = Dif(arr[middle]->GetName(), x.GetName());

        if (dif == 0) { //arr[middle]->GetName() == x.GetName()) {
            return middle;
        }
        else if (dif < 0 ) { //arr[middle]->GetName() < x.GetName()) {
            left = middle + 1;
        }
        else {
            right = middle - 1;
        }

        middle = (left + right) / 2;
    }
    return -1;
}
```

레퍼런스

- https://github.com/ClauText/ClauParser
- SimdJson - https://github.com/lemire/simdjson
- 웹 브라우저에서의 효율적인 병렬 파싱 알고리즘 - http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE06228615&language=ko_KR
- https://github.com/ClauText/ClauText
- https://github.com/ClauText/ClauExplorer
- https://github.com/vztpv/EU3ToVIC2

Q&A

감사합니다