

記号表

記号表(変数表)を使った記憶域の管理

- 宣言された変数の記憶場所などを管理するために、記号表が必要となる。

```
int main()
{
  int abc;
  int cd;
  int efg;
  cd = 10;
  putint(cd);
}
```

記号表

ident="abc"	address=0
ident="cd"	address=1
ident="efg"	address=2

名前(エントリ)..

記号表に登録する内容

ident="abc"	address=0
-------------	-----------

つづり

値を保持する
アドレス

記号表(変数表)を使った記憶域の管理

- 必要なメモリの領域確保。(下記では3変数分必要)
- `cd = 10, putint(cd)`の各文では、`addr(cd)`を知る必要がある。
記号表から`ident`が"cd"である名前(エントリ)を検索する。

```
int main()
{
  int abc;
  int cd;
  int efg;
  cd = 10;
  putint(cd);
}
```

記号表

ident="abc"	address=0
ident="cd"	address=1
ident="efg"	address=2

コード

PUSH 0 3
LDC 0 10
STV 0 1
LDV 0 1
WRI 0 0
POP 0 3
HLT 0 0

記号表(変数表)を使った記憶域の管理(再掲)

- 宣言された変数の記憶場所などを管理するために、記号表が必要となる。

```
int main()
{
  int abc;
  int cd;
  int efg;
  cd = 10;
  putint(cd);
}
```

記号表

ident="abc"	address=0
ident="cd"	address=1
ident="efg"	address=2

名前(エントリ)..

記号表に登録する内容

ident="abc"	address=0
-------------	-----------

つづり

値を保持する
アドレス

宣言(Declaration)

宣言部の文法

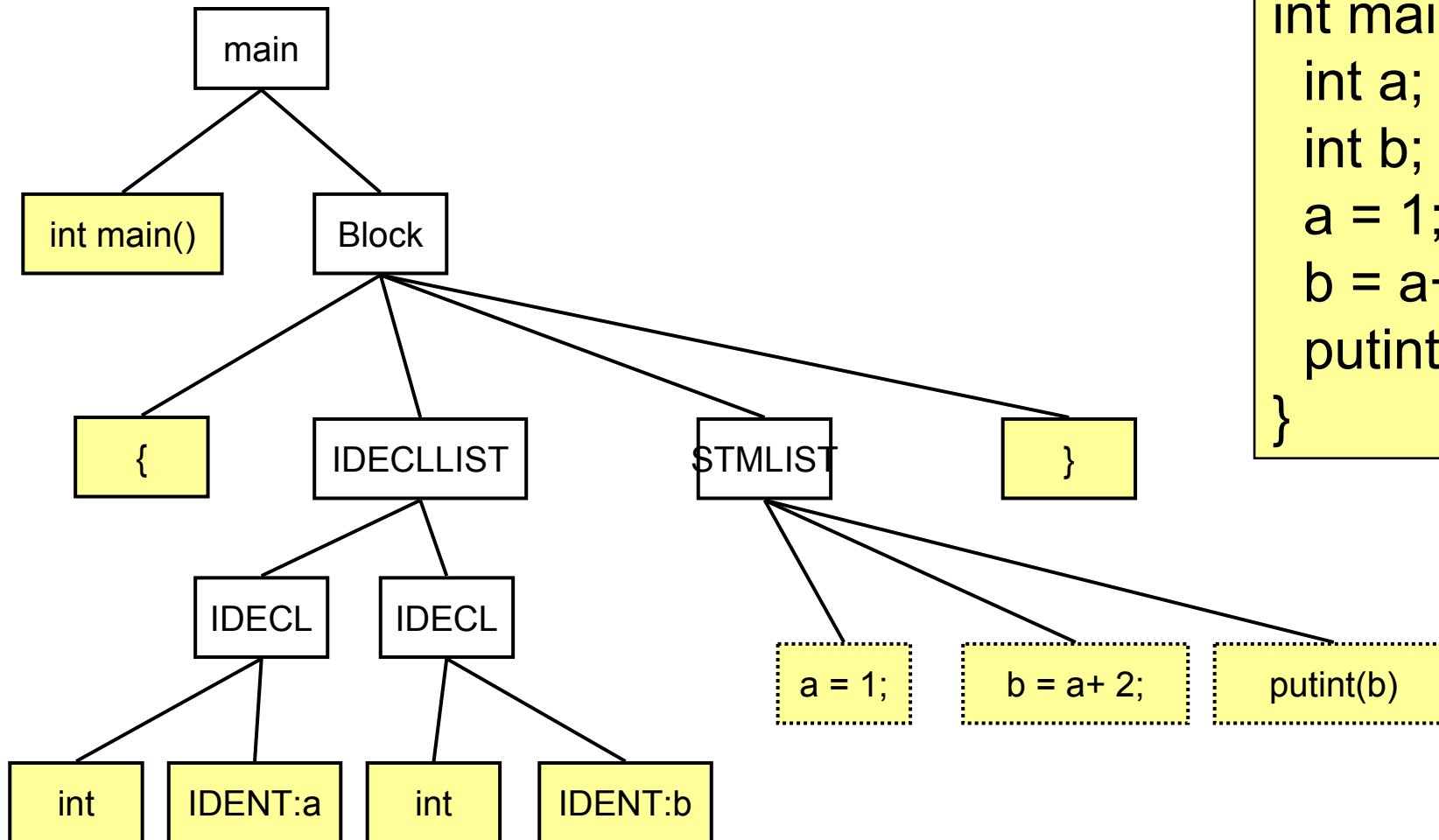
<http://cis.k.hosei.ac.jp/~asasaki/lectureCompiler/problem3.htm>

```
<MAIN> ::= 'int' 'main' '(' ')' <BLOCK>
<BLOCK> ::= '{' <INTDECLLIST> <STATEMENTLIST> '}'
<INTDECLLIST> ::= empty
                | <INTDECLLIST> <INTDECL>
<INTDECL> ::= 'int' <IDENTLIST> ';'
<IDENTLIST> ::= <IDENT>
                | <IDENT> '[' <NUMBER> ']'
                | <IDENTLIST> ',' <IDENT>
                | <IDENTLIST> ',' <IDENT> '[' <NUMBER> ']'
```

```
int a, array[3];
int b;
```

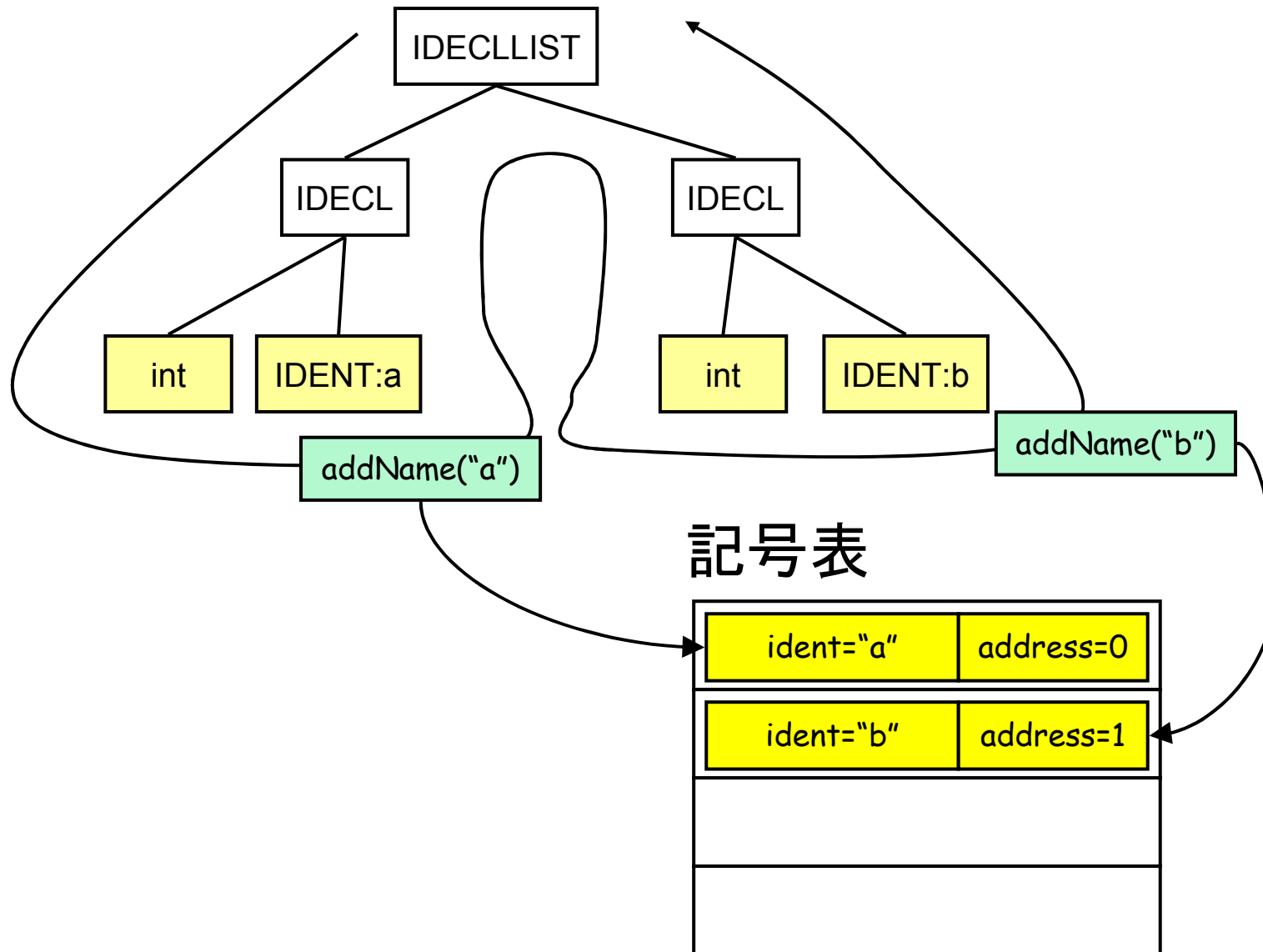
宣言部の例

構文木(概要)



```
int main(){  
  int a;  
  int b;  
  a = 1;  
  b = a+2;  
  putint(b);  
}
```

宣言と記号表への登録



メモリ(記憶域)の確保

