Local Register Allocation

basic idea

- Perform register allocation on a *per basic block* basis
  - Register allocation across basic blocks is *global* — will discuss later
- Perform code generation and register allocation at the same time
  - Find registers for operands when translating 3AC to assembly
- Greedily reuse registers
  - Keep operands in registers if operand is **live**
  - If operand is already in register, no need for new loads
- Only store registers back to the stack if necessary
  - Need register for something else (**spill** register to stack/global memory)
  - At the end of basic block

tracking registers

- As code is generated keep track of:
  - What piece of data is in each register
    - In our case, two possibilities:
      1. Local variable/parameter or global variable
      2. Temporary
  - Whether the data is **dirty** (has its value changed since it was put into the register) — why?

example

```
1:  A = B + C
2:  C = A + B
3:  T1 = B + C
4:  T2 = T1 + C
5:  D = T2
6:  E = A + B
7:  B = E + D
8:  A = C + D
9:  T3 = A + B
10: WRITE(T3)
```

example

```
1:   A = B + C      1:   {A, B}
2:   C = A + B      2:   {A, B, C}
3:   T1 = B + C     3:   {A, B, C, T1}
4:   T2 = T1 + C    4:   {A, B, C, T2}
5:   D = T2         5:   {A, B, C, D}
6:   E = A + B      6:   {C, D, E}
7:   B = E + D      7:   {B, C, D}
8:   A = C + D      8:   {A, B}
9:   T3 = A + B     9:   {T3}
10:  WRITE(T3)      10:  {}
```

Inst	R1	R2	R3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

example

```
1: LW R1 B
   LW R2 C
   ADD R2 R1 R2
```

```
1:  A = B + C      1:  {A, B}
2:  C = A + B      2:  {A, B, C}
3:  T1 = B + C     3:  {A, B, C, T1}
4:  T2 = T1 + C    4:  {A, B, C, T2}
5:  D = T2         5:  {A, B, C, D}
6:  E = A + B      6:  {C, D, E}
7:  B = E + D      7:  {B, C, D}
8:  A = C + D      8:  {A, B}
9:  T3 = A + B     9:  {T3}
10: WRITE(T3)      10: {}
```

Inst	R1	R2	R3
1	B	A*	
2			
3			
4			
5			
6			
7			
8			
9			
10			

example

```
1: LW R1 B
   LW R2 C
   ADD R2 R1 R2
2: ADD R3 R2 R1
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3			
4			
5			
6			
7			
8			
9			
10			

example

```
1: LW R1 B
   LW R2 C
   ADD R2 R1 R2
2: ADD R3 R2 R1
3: ADD R1 R1 R3
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3	T1*	A*	C*
4			
5			
6			
7			
8			
9			
10			

example

```
1: LW R1 B
   LW R2 C
   ADD R2 R1 R2
2: ADD R3 R2 R1
3: ADD R1 R1 R3
4: ADD R1 R1 R3
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3	T1*	A*	C*
4	T2*	A*	C*
5			
6			
7			
8			
9			
10			

example

```
1: LW R1 B
   LW R2 C
   ADD R2 R1 R2
2: ADD R3 R2 R1
3: ADD R1 R1 R3
4: ADD R1 R1 R3
5:
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3	T1*	A*	C*
4	T2*	A*	C*
5	D*	A*	C*
6			
7			
8			
9			
10			

example

```
1: LW R1 B
   LW R2 C
   ADD R2 R1 R2
2: ADD R3 R2 R1
3: ADD R1 R1 R3
4: ADD R1 R1 R3
5:
6: SW R3 C
   LW R3 B
   ADD R2 R2 R3
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3	T1*	A*	C*
4	T2*	A*	C*
5	D*	A*	C*
6	D*	E*	
7			
8			
9			
10			

example

```
1:  LW R1 B
    LW R2 C
    ADD R2 R1 R2
2:  ADD R3 R2 R1
3:  ADD R1 R1 R3
4:  ADD R1 R1 R3
5:
6:  SW R3 C
    LW R3 B
    ADD R2 R2 R3
7:  ADD R2 R2 R1
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3	T1*	A*	C*
4	T2*	A*	C*
5	D*	A*	C*
6	D*	E*	
7	D*	B*	
8			
9			
10			

example

```
1: LW R1 B
   LW R2 C
   ADD R2 R1 R2
2: ADD R3 R2 R1
3: ADD R1 R1 R3
4: ADD R1 R1 R3
5:
6: SW R3 C
   LW R3 B
   ADD R2 R2 R3
7: ADD R2 R2 R1
8: LW C R3
   ADD R1 R3 R1
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3	T1*	A*	C*
4	T2*	A*	C*
5	D*	A*	C*
6	D*	E*	
7	D*	B*	
8	A*	B*	
9			
10			

example

```
1: LW R1 B
   LW R2 C
   ADD R2 R1 R2
2: ADD R3 R2 R1
3: ADD R1 R1 R3
4: ADD R1 R1 R3
5:
6: SW R3 C
   LW R3 B
   ADD R2 R2 R3
7: ADD R2 R2 R1
8: LW C R3
   ADD R1 R3 R1
9: ADD R1 R1 R2
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3	T1*	A*	C*
4	T2*	A*	C*
5	D*	A*	C*
6	D*	E*	
7	D*	B*	
8	A*	B*	
9	T3*		
10			

example

```
1:  LW R1 B
    LW R2 C
    ADD R2 R1 R2
2:  ADD R3 R2 R1
3:  ADD R1 R1 R3
4:  ADD R1 R1 R3
5:
6:  SW R3 C
    LW R3 B
    ADD R2 R2 R3
7:  ADD R2 R2 R1
8:  LW C R3
    ADD R1 R3 R1
9:  ADD R1 R1 R2
10:PUTI R1
```

```
1:   A = B + C
2:   C = A + B
3:   T1 = B + C
4:   T2 = T1 + C
5:   D = T2
6:   E = A + B
7:   B = E + D
8:   A = C + D
9:   T3 = A + B
10:  WRITE(T3)
```

```
1:   {A, B}
2:   {A, B, C}
3:   {A, B, C, T1}
4:   {A, B, C, T2}
5:   {A, B, C, D}
6:   {C, D, E}
7:   {B, C, D}
8:   {A, B}
9:   {T3}
10:  {}
```

Inst	R1	R2	R3
1	B	A*	
2	B	A*	C*
3	T1*	A*	C*
4	T2*	A*	C*
5	D*	A*	C*
6	D*	E*	
7	D*	B*	
8	A*	B*	
9	T3*		
10			

key operations

- **ensure**: make sure that a value exists in a register (put the value in the register, if necessary)
- **allocate**: find a register for a value (kick another value out of a register, if necessary)
- **free**: kick a value out of a register (save the value to the stack/global space if necessary)

key algorithms

```
For each tuple C = A op B in a BB, do
  $R_x$ = ensure(A)
  $R_y$ = ensure(B)
  if A *dead* after this tuple, free($R_x$)
  if B *dead* after this tuple, free($R_y$)
  $R_z$ = allocate(C) //could use $R_x$ or $R_y$
  generate code for op
  mark $R_z$ *dirty*

At end of BB, for each dirty register
  generate code to store register into
appropriate variable
```

```
free(r)
  if r is marked *dirty* and variable is live
    generate store
  mark r as free
```

```
ensure(opr)
  if opr is already in register r
    return r
  else
    r = allocate(opr)
    generate load from opr into r
    return r
```

```
allocate(opr)
  if there is a free r
    choose r
  else
    choose r to free
    free(r)
  mark r associated with opr
  return r
```