

CS323: Compilers

Spring 2023

Week 14:

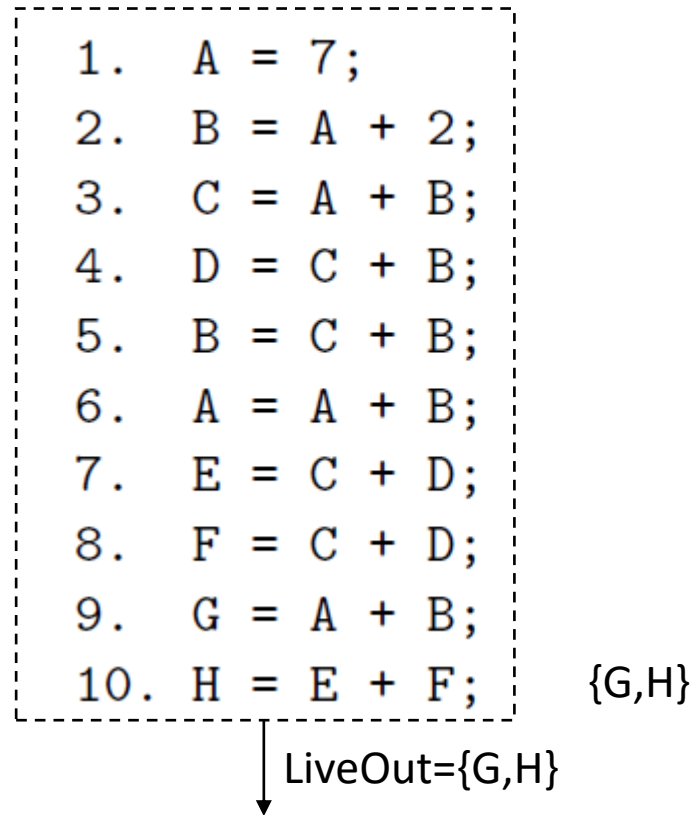
Register allocation via Graph coloring (example), Loop
Dependence Analysis

Register Allocation via Graph Coloring - Example

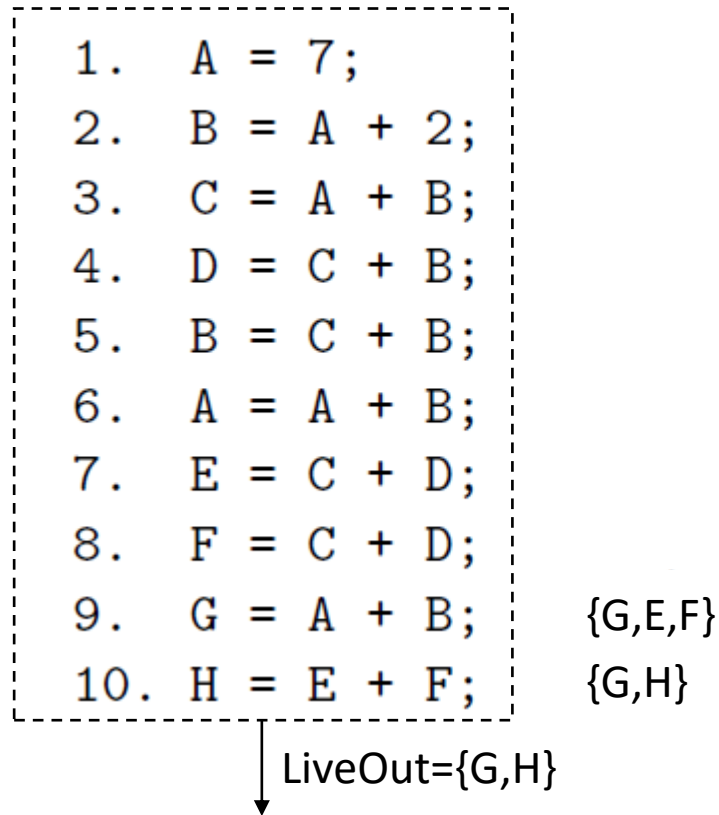
```
1.  A = 7;  
2.  B = A + 2;  
3.  C = A + B;  
4.  D = C + B;  
5.  B = C + B;  
6.  A = A + B;  
7.  E = C + D;  
8.  F = C + D;  
9.  G = A + B;  
10. H = E + F;
```

↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example



Register Allocation via Graph Coloring - Example



Register Allocation via Graph Coloring - Example

```
1.  A = 7;  
2.  B = A + 2;  
3.  C = A + B;  
4.  D = C + B;  
5.  B = C + B;  
6.  A = A + B;  
7.  E = C + D;  
8.  F = C + D;  
9.  G = A + B;  
10. H = E + F;
```

{E, F, A, B}

{G,E,F}

{G,H}

↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example

```
1.  A = 7;  
2.  B = A + 2;  
3.  C = A + B;  
4.  D = C + B;  
5.  B = C + B;  
6.  A = A + B;  
7.  E = C + D;  
8.  F = C + D;  
9.  G = A + B;  
10. H = E + F;
```

{E, A, B, C, D}

{E, F, A, B}

{G,E,F}

{G,H}

↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example

```
1.  A = 7;  
2.  B = A + 2;  
3.  C = A + B;  
4.  D = C + B;  
5.  B = C + B;  
6.  A = A + B;  
7.  E = C + D;  
8.  F = C + D;  
9.  G = A + B;  
10. H = E + F;
```

{A, B, C, D}

{E, A, B, C, D}

{E, F, A, B}

{G,E,F}

{G,H}

↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example

```
1.  A = 7;  
2.  B = A + 2;  
3.  C = A + B;  
4.  D = C + B;  
5.  B = C + B;  
6.  A = A + B;  
7.  E = C + D;  
8.  F = C + D;  
9.  G = A + B;  
10. H = E + F;
```

{A, B, C, D}

{A, B, C, D}

{E, A, B, C, D}

{E, F, A, B}

{G,E,F}

{G,H}

↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example

```
1.  A = 7;  
2.  B = A + 2;  
3.  C = A + B;  
4.  D = C + B;  
5.  B = C + B;  
6.  A = A + B;  
7.  E = C + D;  
8.  F = C + D;  
9.  G = A + B;  
10. H = E + F;
```

{A, B, C, D}

{A, B, C, D}

{A, B, C, D}

{E, A, B, C, D}

{E, F, A, B}

{G,E,F}

{G,H}

↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example

```
1.  A = 7;  
2.  B = A + 2;  
3.  C = A + B;  
4.  D = C + B;  
5.  B = C + B;  
6.  A = A + B;  
7.  E = C + D;  
8.  F = C + D;  
9.  G = A + B;  
10. H = E + F;
```

{A, B, C}

{A, B, C, D}

{A, B, C, D}

{A, B, C, D}

{E, A, B, C, D}

{E, F, A, B}

{G,E,F}

{G,H}

↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example

1. A = 7;	
2. B = A + 2;	{A, B}
3. C = A + B;	{A, B, C}
4. D = C + B;	{A, B, C, D}
5. B = C + B;	{A, B, C, D}
6. A = A + B;	{A, B, C, D}
7. E = C + D;	{E, A, B, C, D}
8. F = C + D;	{E, F, A, B}
9. G = A + B;	{G, E, F}
10. H = E + F;	{G, H}

↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example

1. A = 7;	{A}
2. B = A + 2;	{A, B}
3. C = A + B;	{A, B, C}
4. D = C + B;	{A, B, C, D}
5. B = C + B;	{A, B, C, D}
6. A = A + B;	{A, B, C, D}
7. E = C + D;	{E, A, B, C, D}
8. F = C + D;	{E, F, A, B}
9. G = A + B;	{G, E, F}
10. H = E + F;	{G, H}

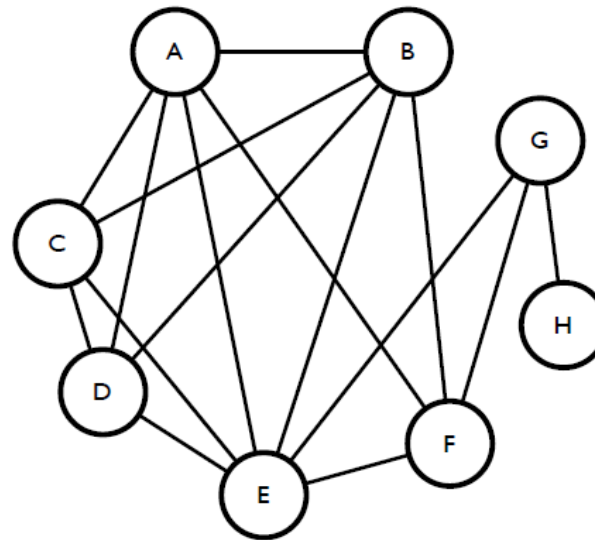
↓ LiveOut={G,H}

Register Allocation via Graph Coloring - Example

{A}
{A, B}
{A, B, C}
{A, B, C, D}
{A, B, C, D}
{A, B, C, D}
{E, A, B, C, D}
{E, F, A, B}

{G,E,F}

{G,H}



Remove H

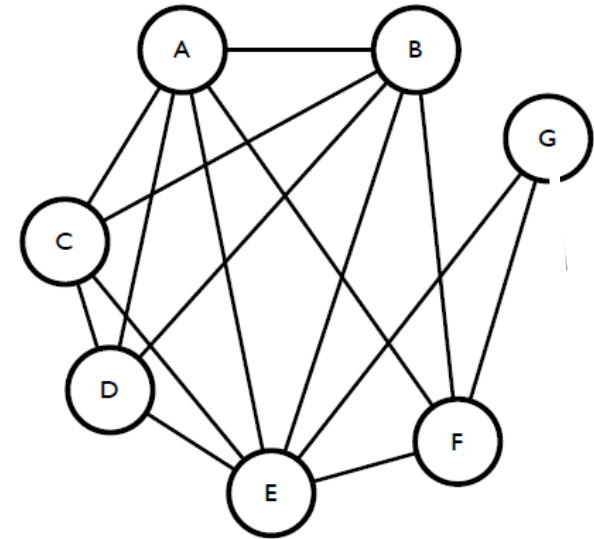
Interference graph

Customized rules (3-coloring):

- Remove nodes in reverse alphabetical order
- Spill variables that are used least (spill the variable with most number of edges in case of a tie)

Register Allocation via Graph Coloring - Example

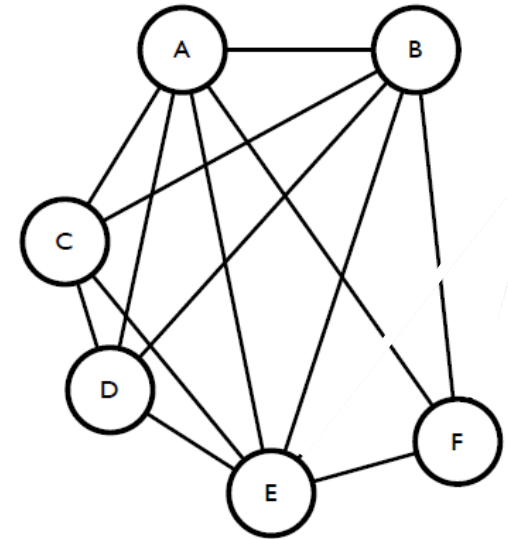
1. $A = 7;$	$\{A\}$
2. $B = A + 2;$	$\{A, B\}$
3. $C = A + B;$	$\{A, B, C\}$
4. $D = C + B;$	$\{A, B, C, D\}$
5. $B = C + B;$	$\{A, B, C, D\}$
6. $A = A + B;$	$\{A, B, C, D\}$
7. $E = C + D;$	$\{E, A, B, C, D\}$
8. $F = C + D;$	$\{E, F, A, B\}$
9. $G = A + B;$	$\{G, E, F\}$
10. $H = E + F;$	$\{G, H\}$



Remove G

Register Allocation via Graph Coloring - Example

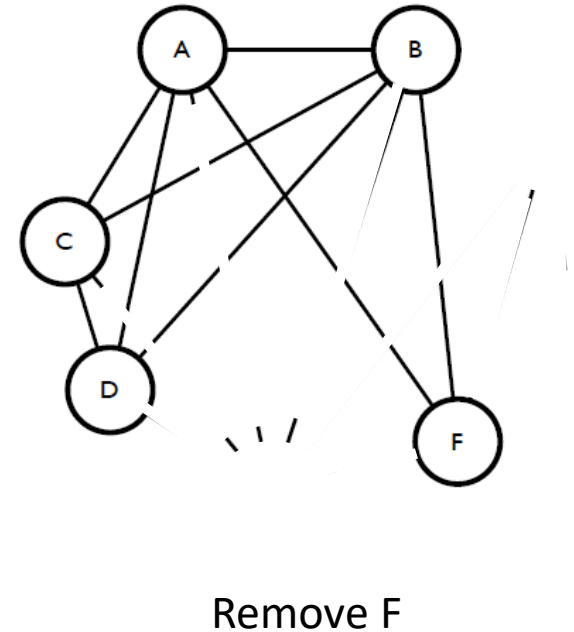
1. $A = 7;$	$\{A\}$
2. $B = A + 2;$	$\{A, B\}$
3. $C = A + B;$	$\{A, B, C\}$
4. $D = C + B;$	$\{A, B, C, D\}$
5. $B = C + B;$	$\{A, B, C, D\}$
6. $A = A + B;$	$\{A, B, C, D\}$
7. $E = C + D;$	$\{E, A, B, C, D\}$
8. $F = C + D;$	$\{E, F, A, B\}$
9. $G = A + B;$	$\{G, E, F\}$
10. $H = E + F;$	$\{G, H\}$



Remove E

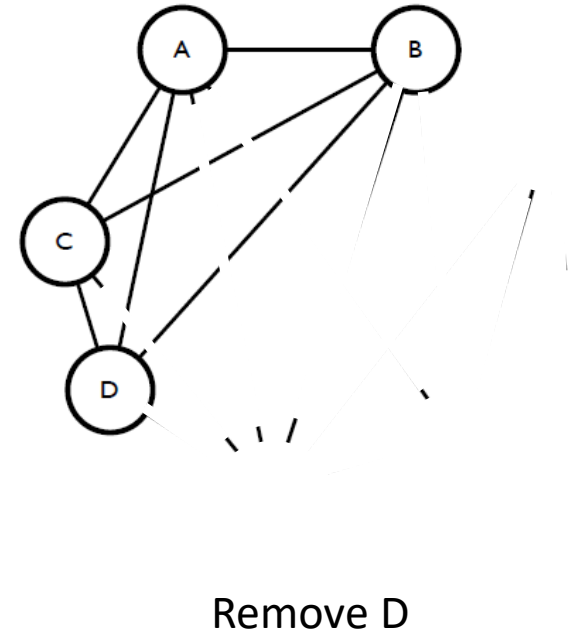
Register Allocation via Graph Coloring - Example

1. $A = 7;$	$\{A\}$
2. $B = A + 2;$	$\{A, B\}$
3. $C = A + B;$	$\{A, B, C\}$
4. $D = C + B;$	$\{A, B, C, D\}$
5. $B = C + B;$	$\{A, B, C, D\}$
6. $A = A + B;$	$\{A, B, C, D\}$
7. $E = C + D;$	$\{E, A, B, C, D\}$
8. $F = C + D;$	$\{E, F, A, B\}$
9. $G = A + B;$	$\{G, E, F\}$
10. $H = E + F;$	$\{G, H\}$



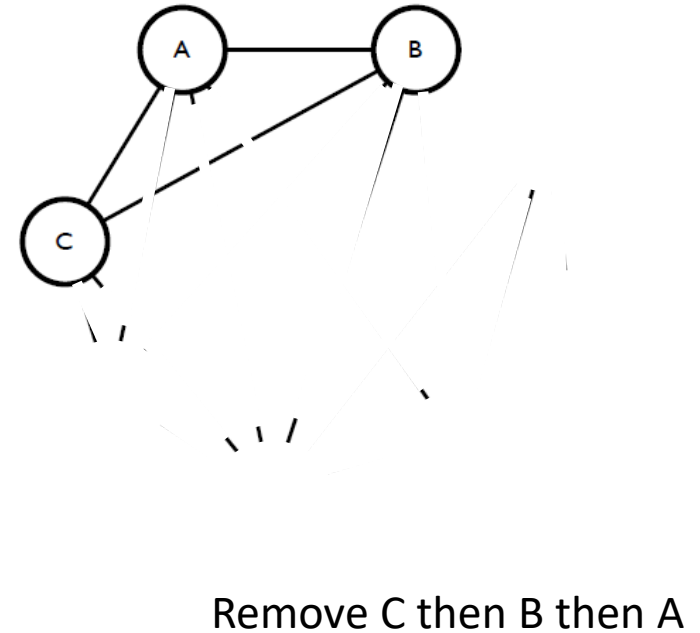
Register Allocation via Graph Coloring - Example

1. $A = 7;$	$\{A\}$
2. $B = A + 2;$	$\{A, B\}$
3. $C = A + B;$	$\{A, B, C\}$
4. $D = C + B;$	$\{A, B, C, D\}$
5. $B = C + B;$	$\{A, B, C, D\}$
6. $A = A + B;$	$\{A, B, C, D\}$
7. $E = C + D;$	$\{E, A, B, C, D\}$
8. $F = C + D;$	$\{E, F, A, B\}$
9. $G = A + B;$	$\{G, E, F\}$
10. $H = E + F;$	$\{G, H\}$



Register Allocation via Graph Coloring - Example

1. $A = 7;$	$\{A\}$
2. $B = A + 2;$	$\{A, B\}$
3. $C = A + B;$	$\{A, B, C\}$
4. $D = C + B;$	$\{A, B, C, D\}$
5. $B = C + B;$	$\{A, B, C, D\}$
6. $A = A + B;$	$\{A, B, C, D\}$
7. $E = C + D;$	$\{E, A, B, C, D\}$
8. $F = C + D;$	$\{E, F, A, B\}$
9. $G = A + B;$	$\{G, E, F\}$
10. $H = E + F;$	$\{G, H\}$



Register Allocation via Graph Coloring - Example

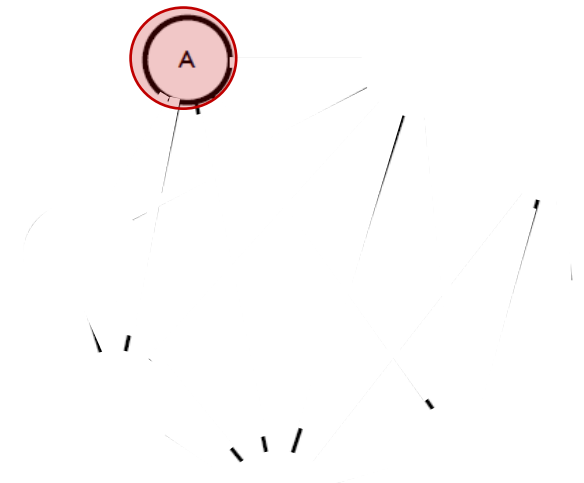
1.	A = 7;	{A}
2.	B = A + 2;	{A, B}
3.	C = A + B;	{A, B, C}
4.	D = C + B;	{A, B, C, D}
5.	B = C + B;	{A, B, C, D}
6.	A = A + B;	{A, B, C, D}
7.	E = C + D;	{E, A, B, C, D}
8.	F = C + D;	{E, F, A, B}
9.	G = A + B;	{G, E, F}
10.	H = E + F;	{G, H}

Stack: A
B
C
D
F
E
G
H

Register Allocation via Graph Coloring - Example

1. $A = 7;$	$\{A\}$
2. $B = A + 2;$	$\{A, B\}$
3. $C = A + B;$	$\{A, B, C\}$
4. $D = C + B;$	$\{A, B, C, D\}$
5. $B = C + B;$	$\{A, B, C, D\}$
6. $A = A + B;$	$\{A, B, C, D\}$
7. $E = C + D;$	$\{E, A, B, C, D\}$
8. $F = C + D;$	$\{E, F, A, B\}$
9. $G = A + B;$	$\{G, E, F\}$
10. $H = E + F;$	$\{G, H\}$

3-Color the variables:

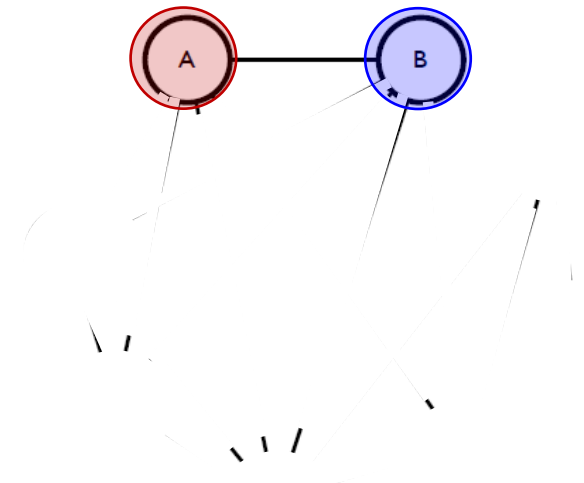


Stack: A - Red
 B
 C
 D
 F
 E
 G
 H

Register Allocation via Graph Coloring - Example

1.	A = 7;	{A}
2.	B = A + 2;	{A, B}
3.	C = A + B;	{A, B, C}
4.	D = C + B;	{A, B, C, D}
5.	B = C + B;	{A, B, C, D}
6.	A = A + B;	{A, B, C, D}
7.	E = C + D;	{E, A, B, C, D}
8.	F = C + D;	{E, F, A, B}
9.	G = A + B;	{G, E, F}
10.	H = E + F;	{G, H}

3-Color the variables:



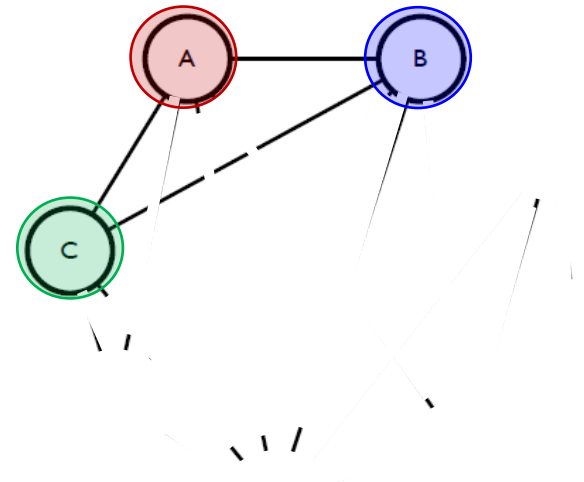
Stack:

B - Blue
C
D
F
E
G
H

Register Allocation via Graph Coloring - Example

1. $A = 7;$	$\{A\}$
2. $B = A + 2;$	$\{A, B\}$
3. $C = A + B;$	$\{A, B, C\}$
4. $D = C + B;$	$\{A, B, C, D\}$
5. $B = C + B;$	$\{A, B, C, D\}$
6. $A = A + B;$	$\{A, B, C, D\}$
7. $E = C + D;$	$\{E, A, B, C, D\}$
8. $F = C + D;$	$\{E, F, A, B\}$
9. $G = A + B;$	$\{G, E, F\}$
10. $H = E + F;$	$\{G, H\}$

3-Color the variables:



Stack:

C-Green

D

F

E

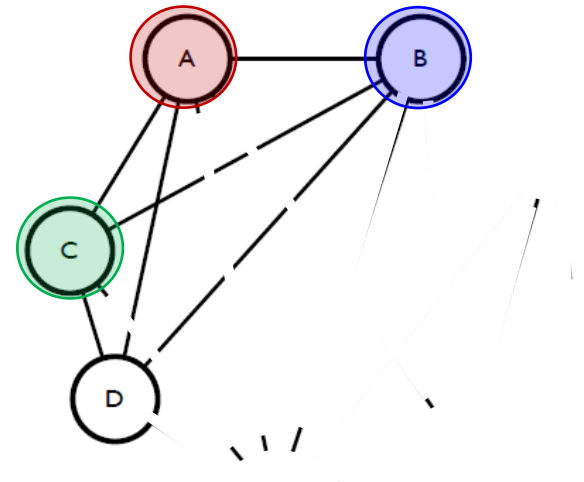
G

H

Register Allocation via Graph Coloring - Example

1. $A = 7;$	$\{A\}$
2. $B = A + 2;$	$\{A, B\}$
3. $C = A + B;$	$\{A, B, C\}$
4. $D = C + B;$	$\{A, B, C, D\}$
5. $B = C + B;$	$\{A, B, C, D\}$
6. $A = A + B;$	$\{A, B, C, D\}$
7. $E = C + D;$	$\{E, A, B, C, D\}$
8. $F = C + D;$	$\{E, F, A, B\}$
9. $G = A + B;$	$\{G, E, F\}$
10. $H = E + F;$	$\{G, H\}$

3-Color the variables: Spill D



Stack:

D - ??
F
E
G
H

Register Allocation via Graph Coloring - Example

Earlier code:

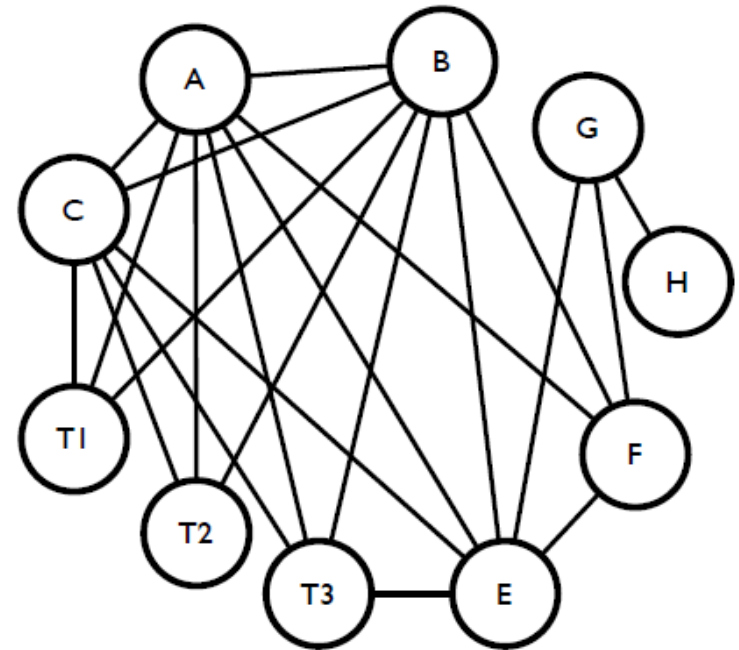
```
A = 7;
B = A + 2;
C = A + B;
D = C + B;
B = C + B;
A = A + B;
E = C + D;
F = C + D;
G = A + B;
H = E + F;
```

Rewritten code:

```
1. A = 7;
2. B = A + 2;
3. C = A + B;
4. T1 = C + B;
4'. ST T1, D
5. B = C + B;
6. A = A + B;
6'. LD D, T2
7. E = C + T2;
7'. LD D, T3
8. F = C + T3;
9. G = A + B;
10. H = E + F;
```

Liveness info:

```
{A}
{A, B}
{A, B, C}
{A, B, C, T1}
{A, B, C}
{A, B, C}
{A, B, C, T2}
{A, B, C, E}
{A, B, C, E, T3}
{A, B, E, F}
{G, E, F}
{G, H}
```

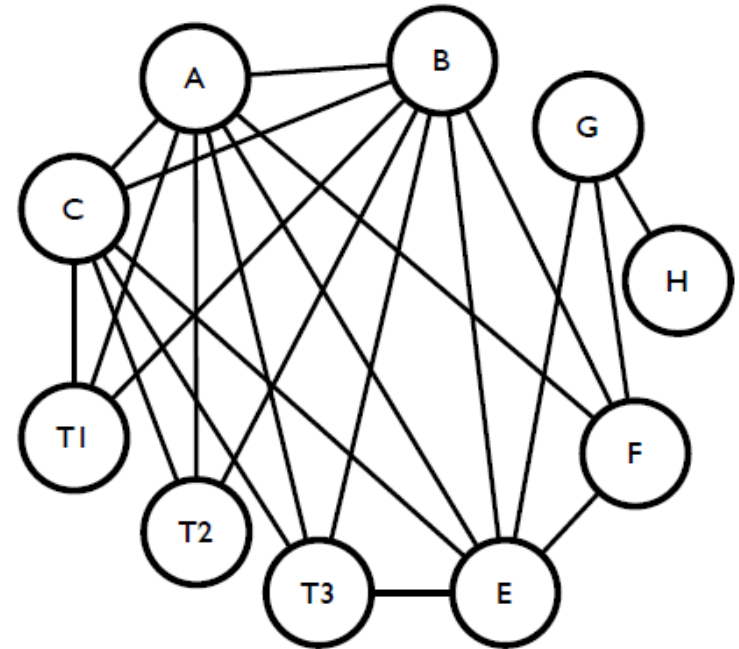


New interference graph

3-Color the variables: Spill D, rewrite code and recalculate liveness

Register Allocation via Graph Coloring - Example

1. A = 7;	{A}
2. B = A + 2;	{A, B}
3. C = A + B;	{A, B, C}
4. T1 = C + B;	{A, B, C, T1}
4'. ST T1, D	{A, B, C}
5. B = C + B;	{A, B, C}
6. A = A + B;	{A, B, C}
6'. LD D, T2	{A, B, C, T2}
7. E = C + T2;	{A, B, C, E}
7'. LD D, T3	{A, B, C, E, T3}
8. F = C + T3;	{A, B, E, F}
9. G = A + B;	{G, E, F}
10. H = E + F;	{G, H}

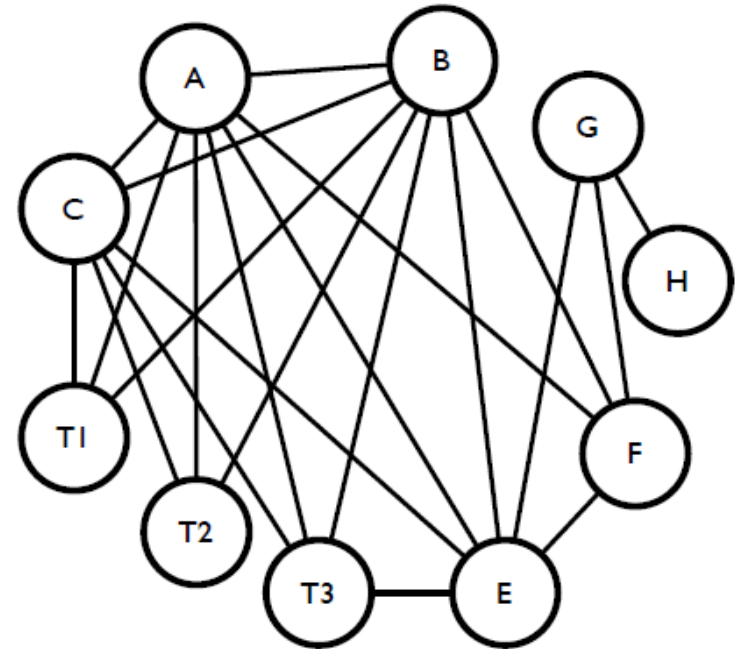


Simplify (step 1)

Stack (left-bottom, right-top): H, G, E, F, C, T1, T2, T3, B, A

Register Allocation via Graph Coloring - Example

1. A = 7;	{A}
2. B = A + 2;	{A, B}
3. C = A + B;	{A, B, C}
4. T1 = C + B;	{A, B, C, T1}
4'. ST T1, D	{A, B, C}
5. B = C + B;	{A, B, C}
6. A = A + B;	{A, B, C}
6'. LD D, T2	{A, B, C, T2}
7. E = C + T2;	{A, B, C, E}
7'. LD D, T3	{A, B, C, E, T3}
8. F = C + T3;	{A, B, E, F}
9. G = A + B;	{G, E, F}
10. H = E + F;	{G, H}

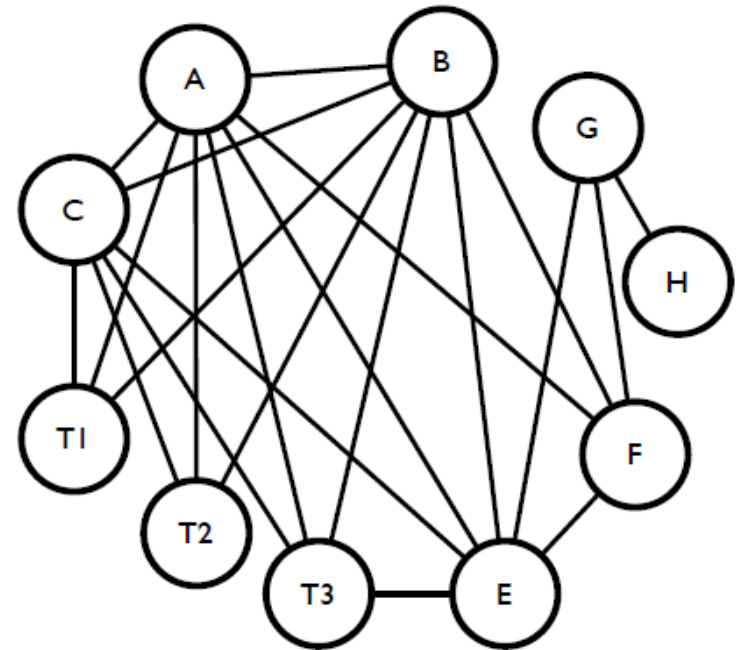


Color (step 2) Stack (left-bottom, right-top): H, G, E, F, C, T1, T2, T3, B, A

Which node must be Spilled now?

Register Allocation via Graph Coloring - Example

1. A = 7;	{A}
2. B = A + 2;	{A, B}
3. C = A + B;	{A, B, C}
4. T1 = C + B;	{A, B, C, T1}
4'. ST T1, D	{A, B, C}
5. B = C + B;	{A, B, C}
6. A = A + B;	{A, B, C}
6'. LD D, T2	{A, B, C, T2}
7. E = C + T2;	{A, B, C, E}
7'. LD D, T3	{A, B, C, E, T3}
8. F = C + T3;	{A, B, E, F}
9. G = A + B;	{G, E, F}
10. H = E + F;	{G, H}

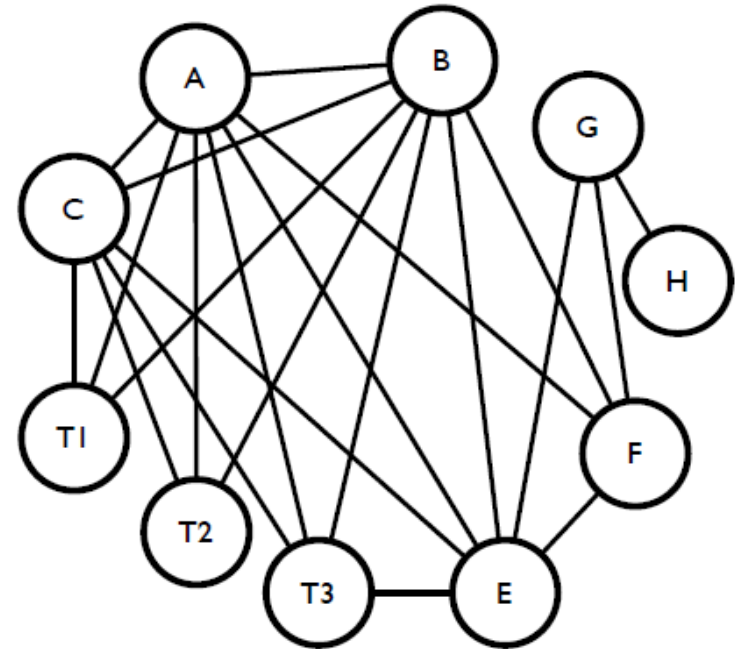


Color (step 2) Stack (left-bottom, right-top): H, G, E, F, C, T1, T2, T3, B, A

Which node must be Spilled now? (i.e. which node can't be colored?)

Register Allocation via Graph Coloring - Example

1. A = 7;	{A}
2. B = A + 2;	{A, B}
3. C = A + B;	{A, B, C}
4. T1 = C + B;	{A, B, C, T1}
4'. ST T1, D	{A, B, C}
5. B = C + B;	{A, B, C}
6. A = A + B;	{A, B, C}
6'. LD D, T2	{A, B, C, T2}
7. E = C + T2;	{A, B, C, E}
7'. LD D, T3	{A, B, C, E, T3}
8. F = C + T3;	{A, B, E, F}
9. G = A + B;	{G, E, F}
10. H = E + F;	{G, H}



Color (step 2)

Stack (left-bottom, right-top): H, G, E, F, C, T1, T2, T3, B, A

Which node must be Spilled now? (C. Now repeat the steps starting from rewriting the code to spill C, calculating liveness, drawing iteration graph and then simplifying the iteration graph.)

Overall Algorithm

