## 15-411: Calling Conventions

Jan Hoffmann

### Example: Recursive Pow

```
int pow(int b, int e)
//@requires e >= 0;
{
    if (e == 0)
        return 1;
    else
        return b * pow(b, e-1);
}
```

```
pow(b,e):
    if (e == 0) then done else recurse
done:
    ret 1
recurse:
    t0 <- e - 1
    t1 <- pow(b, t0)
    t2 <- b * t1
    ret t2</pre>
```

# C0 int pow(int b, int e) //@requires e >= 0; { if (e == 0) return 1; else return b \* pow(b, e-1); }

pow(b,e):
 if (e == 0) then done else recurse
done:
 ret 1
recurse:
 t0 <- e - 1
 t1 <- pow(b, t0)
 t2 <- b \* t1
 ret t2</pre>

## C0 int pow(int b, int e) //@requires e >= 0; { if (e == 0) return 1; else return b \* pow(b, e-1); }

### 3-Address code.

```
pow(b,e):
    if (e == 0) then done else recurse
done:
    ret 1
recurse:
    t0 <- e - 1
    t1 <- pow(b, t0)
    t2 <- b * t1
    ret t2</pre>
```

program	def	use	live-in
pow(b,e) :			
if $(e == 0)$ then done else recurse			
done :			
ret 1			
recurse :			
$t_0 \leftarrow e - 1$			
$t_1 \leftarrow pow(b, t_0)$			
$t_2 \leftarrow b * t_1$			
ret $t_2$			

program	def	use	live-in
pow(b,e):	b, e		
if $(e == 0)$ then done else recurse			
done :			
ret 1			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$		
$t_1 \leftarrow pow(b, t_0)$	$t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
ret $t_2$			

program	def	use	live-in
pow(b,e) :	b, e		
if $(e == 0)$ then done else recurse		e	
done :			
ret 1			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$	e	
$t_1 \leftarrow pow(b, t_0)$	$t_1$	$b, t_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
ret $t_2$		$t_2$	

program	def	use	live-in
pow(b,e) :	b, e		
if $(e == 0)$ then done else recurse		e	
done :			
ret 1			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$	e	
$t_1 \leftarrow pow(b, t_0)$	$t_1$	$b, t_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
ret $t_2$		$t_2$	$t_2$

program	def	use	live-in
pow(b,e):	b, e		
if $(e == 0)$ then done else recurse		e	
done :			
ret 1			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$	e	
$t_1 \leftarrow pow(b, t_0)$	$t_1$	$b, t_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
ret $t_2$		$t_2$	$t_2$

program	def	use	live-in
pow(b,e) :	b, e		
if $(e == 0)$ then done else recurse		e	
done :			
ret 1			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$	e	
$t_1 \leftarrow pow(b, t_0)$	$t_1$	$b, t_0$	$b, t_0$ $b, t_1$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
ret $t_2$		$t_2$	$t_2$

program	def	use	live-in
pow(b,e):	b, e		
if $(e == 0)$ then done else recurse		e	
done :			
ret 1			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$	e	b, e
$t_1 \leftarrow pow(b, t_0)$	$t_1$	$e \\ b, t_0$	$b, t_0$
$t_2 \leftarrow b * t_1$	$t_2$		$b, t_1$
ret $t_2$		$t_2$	$t_2$

program	def	use	live-in
pow(b,e) :	b, e		
if $(e == 0)$ then done else recurse		e	
done :			
ret 1			
recurse :			b, e
$t_0 \leftarrow e - 1$	$t_0$	e	$egin{array}{c} b,e\ b,e \end{array}$
$t_1 \leftarrow pow(b, t_0)$	$t_1$	$b, t_0$	$b, t_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
ret $t_2$		$t_2$	$t_2$

program	def	use	live-in
pow(b,e) :	b, e		
if $(e == 0)$ then done else recurse		e	
done :			
ret 1			
recurse :			b, e
$t_0 \leftarrow e - 1$	$t_0$	e	$egin{array}{c} b,e\ b,e \end{array}$
$t_1 \leftarrow pow(b, t_0)$	$t_1$	$b, t_0$	$b, t_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
ret $t_2$		$t_2$	$t_2$

program	def	use	live-in
pow(b,e) :	b, e		
if $(e == 0)$ then done else recurse		e	
done :			
ret 1			
recurse :			b, e
$t_0 \leftarrow e - 1$	$t_0$	e	$egin{array}{c} b,e\ b,e \end{array}$
$t_1 \leftarrow pow(b, t_0)$	$t_1$	$b, t_0$	$b, t_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
ret $t_2$		$t_2$	$t_2$

program	def	use	live-in
pow(b,e) :	b, e		
if $(e == 0)$ then done else recurse		e	b, e
done :			
ret 1			
recurse :			b, e
$t_0 \leftarrow e - 1$	$t_0$	e	b, e
$t_1 \leftarrow pow(b, t_0)$	$t_1$		$b, t_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
ret $t_2$		$t_2$	$t_2$

### Rules for Low-Level 3-Address Code

$$\begin{array}{c} l: \mathsf{call} \ f\\ \mathsf{caller}\text{-}\mathsf{save}(r)\\ \hline \\ \mathsf{def}(l,r) \end{array} J_8' \end{array}$$

$$\begin{array}{c} l: \mathsf{ret} \ s \\ \mathsf{callee}\text{-}\mathsf{save}(r) \\ \hline \\ \mathsf{use}(l,r) \end{array} \ J_2' \end{array}$$

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$			
$arg_2 \leftarrow t_0$			
$arg_1 \leftarrow b$			
call pow			
$t_1 \leftarrow res_0$			
$t_2 \leftarrow b * t_1$			
$res_0 \leftarrow t_2$			
ret			
l in a	reness	Take into a	account pre-colored
	01000	registers	
		-	

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$			
$arg_2 \leftarrow t_0$			
$arg_1 \leftarrow b$			
call pow			
$t_1 \leftarrow res_0$			
$t_2 \leftarrow b * t_1$			
$res_0 \leftarrow t_2$			
ret			
l in a	reness	Take into a	account pre-colored
	01000	registers	
		-	

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$			
$arg_2 \leftarrow t_0$			
$arg_1 \leftarrow b$			
call pow			
$t_1 \leftarrow res_0$			
$t_2 \leftarrow b * t_1$			
$res_0 \leftarrow t_2$	$res_0$		
ret			
L iv	reness		ccount pre-colore
		registers	

		live-in
$t_2$		
$res_0$		
		ccount pre-colore
	res <sub>0</sub>	

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$			
$arg_2 \leftarrow t_0$			
$arg_1 \leftarrow b$			
call pow			
$t_1 \leftarrow res_0$	$t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			
		·	
Liv	reness	Take into a registers	ccount pre-colore

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$			
$arg_2 \leftarrow t_0$			
$arg_1 \leftarrow b$			
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$			
$arg_2 \leftarrow t_0$			
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$			
$arg_2 \leftarrow t_0$	$arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$\left \begin{array}{c} arg_3, arg_4, arg_5, \\ arg_6, ler_7, ler_8 \end{array}\right $		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$		
$arg_2 \leftarrow t_0$	$arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$  arg_3, arg_4, arg_5,$		
	$\begin{array}{c c} res_0, arg_1, arg_2,\\ arg_3, arg_4, arg_5,\\ arg_6, ler_7, ler_8 \end{array}$		
$t_1 \leftarrow res_0$	$ t_1 $		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$  arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$  res_0, arg_1, arg_2,$		
	$  arg_3, arg_4, arg_5,$		
	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$  t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$			
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$  arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$  res_0, arg_1, arg_2,$		
	$  arg_3, arg_4, arg_5,$		
	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$  t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$  arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$  arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$  arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$  t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :			
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$  arg_3, arg_4, arg_5,$		
	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$  arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$ \begin{array}{c c} res_0, arg_1, arg_2, \\ arg_3, arg_4, arg_5, \\ arg_6, ler_7, ler_8 \end{array} $		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret			

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$		
$arg_2 \leftarrow t_0$	$arg_2$		
$arg_1 \leftarrow b$	$arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$t_1$		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$		
ret		$res_0$	
Liv		āke into a egisters	ccount pre-colore

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$		
$arg_2 \leftarrow t_0$	$arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $		
$t_2 \leftarrow b * t_1$	$t_2$		
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	
Liv		āke into a egisters	ccount pre-color

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$		
$t_2 \leftarrow b * t_1$	$  t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$  arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$		
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$  t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	$\mid b$		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$arg_2$		
$arg_1 \leftarrow b$	$  arg_1$		
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$  t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$		
$arg_2 \leftarrow t_0$	$arg_2$		
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$t_0$		
$arg_2 \leftarrow t_0$	$arg_2$	$ t_0 $	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

registers

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$  res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret			
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$  res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$ t_0 $	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$ t_0 $	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e		
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$ t_0 $	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b		
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$arg_2$	$t_0$	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$\mid t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$  arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$arg_2$	$t_0$	
$arg_1 \leftarrow b$	$arg_1$	b	
call pow	$  res_0, arg_1, arg_1$	$g_2$ , $  arg_1, arg_2  $	
	$  arg_3, arg_4, arg_4$	$g_{5},$	
	$  arg_6, ler_7, ler$	8	
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	
ret		$res_0$	$res_0$
		1	1
Liv	reness	Take into a registers	ccount pre-colored

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$  arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$t_0$	e	
$arg_2 \leftarrow t_0$	$arg_2$	$t_0$	
$arg_1 \leftarrow b$	$arg_1$	b	
call pow	$  res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$  arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$
			·
L iv.	veness	Take into a	ccount pre-col
		registers	

registers

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$  arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$  t_0$	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$ \begin{array}{c c} arg_3, arg_4, arg_5, \\ arg_6, ler_7, ler_8 \end{array} $		
$t_1 \leftarrow res_0$	$  t_1$	$res_0$	
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	
$arg_1 \leftarrow b$	$  arg_1$	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$t_0$	e	
$arg_2 \leftarrow t_0$	$arg_2$	$t_0$	
$arg_1 \leftarrow b$	$  arg_1  $	b	
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$\begin{array}{c} res_0, arg_1, arg_2,\\ arg_3, arg_4, arg_5,\\ arg_6, ler_7, ler_8 \end{array}$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$t_1$	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$arg_3, arg_4, arg_5,$		
	$ \begin{array}{c} arg_3, arg_4, arg_5, \\ arg_6, ler_7, ler_8 \end{array} $		
$t_1 \leftarrow res_0$	$t_1$	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	
$arg_2 \leftarrow t_0$	$arg_2$	$t_0$	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$t_1$	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			
$t_0 \leftarrow e - 1$	$  t_0$	e	b, e
$arg_2 \leftarrow t_0$	$arg_2$	$t_0$	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1  $	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$  arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$  arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	
recurse :			b, e
$t_0 \leftarrow e - 1$	$  t_0$	e	b, e
$arg_2 \leftarrow t_0$	$arg_2$	$ t_0 $	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	$res_0$
recurse :			b, e
$t_0 \leftarrow e - 1$	$t_0$	e	b, e
$arg_2 \leftarrow t_0$	$arg_2$	$ t_0 $	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	$res_0$
recurse :			b, e
$t_0 \leftarrow e - 1$	$t_0$	e	b, e
$arg_2 \leftarrow t_0$	$arg_2$	$ t_0 $	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	
if $(e == 0)$ then done else recurse			b, e
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	$res_0$
recurse :			b, e
$t_0 \leftarrow e - 1$	$  t_0$	e	b, e
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$  arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	
$e \leftarrow arg_2$	e	$arg_2$	$b, arg_2$
if $(e == 0)$ then done else recurse			b, e
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	$res_0$
recurse :			b, e
$t_0 \leftarrow e - 1$	$  t_0$	e	b, e
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$  arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	$arg_1, arg_2$
$e \leftarrow arg_2$	e	$arg_2$	$b, arg_2$
if $(e == 0)$ then done else recurse			b, e
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	$res_0$
recurse :			b, e
$t_0 \leftarrow e - 1$	$  t_0$	e	b, e
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

program	def	use	live-in
pow :	$arg_1, arg_2$		
$b \leftarrow arg_1$	b	$arg_1$	$arg_1, arg_2$
$e \leftarrow arg_2$	e	$arg_2$	$b, arg_2$
if $(e == 0)$ then done else recurse			b, e
done :			
$res_0 \leftarrow 1$	$res_0$		
ret		$res_0$	$res_0$
recurse :			b, e
$t_0 \leftarrow e - 1$	$t_0$	e	b, e
$arg_2 \leftarrow t_0$	$  arg_2$	$t_0$	$b, t_0$
$arg_1 \leftarrow b$	$  arg_1$	b	$b, arg_2$
call pow	$res_0, arg_1, arg_2,$	$arg_1, arg_2$	$b, arg_1, arg_2$
	$arg_3, arg_4, arg_5,$		
	$arg_6, ler_7, ler_8$		
$t_1 \leftarrow res_0$	$ t_1 $	$res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$t_2$	$b, t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$res_0$	$t_2$	$t_2$
ret		$res_0$	$res_0$

temp	interfering with
b	$[res_0, arg_1, arg_2, arg_3, arg_4, arg_5, arg_6, ler_7, ler_8, e, t_0, t_1]$
e	b
$t_0$	b
$t_1$	b
$t_2$	

### Interference Graph

All pre-colored registers interfere with each other

No caller-saved register for b available. Use lee9.

temp	interfering with
b	$res_0, arg_1, arg_2, arg_3, arg_4, arg_5, arg_6, ler_7, ler_8, e, t_0, t_1$
e	b
$t_0$	b
$t_1$	b
$t_2$	

### Interference Graph

All pre-colored registers interfere with each other

pow :	$arg_1, arg_2, lee_9$
	ana ana loo
push $lee_9$	$arg_1, arg_2, lee_9$
$b \leftarrow arg_1$	$arg_1, arg_2$
$e \leftarrow arg_2$	$b, arg_2$
if $(e == 0)$ then done else recurse	b, e
done :	
$res_0 \leftarrow 1$	
goto exitpow	$res_0$
recurse :	b, e
$t_0 \leftarrow e - 1$	b, e
$arg_2 \leftarrow t_0$	$b, t_0$
$arg_1 \leftarrow b$	$b, arg_2$
call pow	$b, arg_1, arg_2$
$t_1 \leftarrow res_0$	$b, res_0$
$t_2 \leftarrow b * t_1$	$b, t_1$
$res_0 \leftarrow t_2$	$t_2$
goto exitpow	$res_0$
exitpow :	$res_0$
pop lee9	$res_0$
ret	$lee_9, res_0$

# Save Callee-Saved Regs and epilog

program	live-in	
pow :	$arg_1, arg_2, lee_9$	-
push $lee_9$	$arg_1, arg_2, lee_9$	
$b \leftarrow arg_1$	$arg_1, arg_2$	
$e \leftarrow arg_2$	$b, arg_2$	
if $(e == 0)$ then done else recurse	b, e	
done :		Register Allocation:
$res_0 \leftarrow 1$		0
goto exitpow	$res_0$	We order the colors (machine registers) as
recurse :	b, e	$res_0, arg_1, \ldots, arg_6, ler_7, ler_8, lee_9$
$t_0 \leftarrow e - 1$	b, e	
$arg_2 \leftarrow t_0$	$b, t_0$	
$arg_1 \leftarrow b$	$b, arg_2$	
call pow	$b, arg_1, arg_2$	From this we construct the assignment
$t_1 \leftarrow res_0$	$b, res_0$	$b \mapsto lee_9$
$t_2 \leftarrow b * t_1$	$b, t_1$	$e \mapsto res_0$
$res_0 \leftarrow t_2$	$t_2$	$t_0 \mapsto res_0$
goto exitpow	$res_0$	$t_1 \mapsto res_0$
exitpow :	$res_0$	$t_2 \mapsto res_0$
pop lee9	$res_0$	
ret	$lee_9, res_0$	

# Save Callee-Saved Regs and epilog

pow:

push *lee*<sub>9</sub>  $lee_9 \leftarrow arg_1$  $res_0 \leftarrow arg_2$ if  $(res_0 == 0)$  then done else recurse done :  $res_0 \leftarrow 1$ goto exitpow recurse :  $res_0 \leftarrow res_0 - 1$  $arg_2 \leftarrow res_0$  $arg_1 \leftarrow lee_9$ call pow  $res_0 \leftarrow res_0$  $res_0 \leftarrow lee_9 * res_0$  $res_0 \leftarrow res_0$ goto exitpow exitpow : pop  $lee_9$ ret

Optimization and Code Generation

pow: push *lee*<sub>9</sub>  $lee_9 \leftarrow arg_1$  $res_0 \leftarrow arg_2$ if  $(res_0 == 0)$  then done else recurse done :  $res_0 \leftarrow 1$ goto exitpow recurse :  $res_0 \leftarrow res_0 - 1$  $arg_2 \leftarrow res_0$  $arg_1 \leftarrow lee_9$ (redundant) call pow (redundant)  $res_0 \leftarrow res_0$  $res_0 \leftarrow lee_9 * res_0$  $res_0 \leftarrow res_0$ (redundant) goto exitpow exitpow : pop  $lee_9$ ret

### Optimization and Code Generation

pow:

push $lee_9$				
$lee_9 \leftarrow arg_1$				
$res_0 \leftarrow arg_2$				
if $(res_0 == 0)$ then c	lone else recurse	pow:	pushq	%rbx
done :		pow.	movl	%edi, %ebx
$res_0 \leftarrow 1$				•
goto exitpow			movl	,
recurse :			-	\$0, %eax
$res_0 \leftarrow res_0 - 1$			jne L1	±
$arg_2 \leftarrow res_0$			movl	\$1, %eax
$arg_1 \leftarrow lee_9$	(redundant)		goto L2	
call pow		L1:	subl	\$1, %eax
$res_0 \leftarrow res_0$	(redundant)		movl	%eax, %esi
$res_0 \leftarrow lee_9 * res_0$			call	pow
$res_0 \leftarrow res_0$	(redundant)		imull	%ebx, %eax
goto exitpow		L2:	popq %r	·bx
exitpow :			ret	
pop $lee_9$				
ret				

Optimization and Code Generation