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);
}
```

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
```

Recursive Pow

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
```

Recursive Pow

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
```

Recursive Pow

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			

Liveness Analysis (3-address code)

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			

Liveness Analysis (3-address code)

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	

Liveness Analysis (3-address code)

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

Liveness Analysis (3-address code)

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

Liveness Analysis (3-address code)

Liveness Analysis (3-address code)

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
$t_2 \leftarrow b * t_1$	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	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

Liveness Analysis (3-address code)

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	b, e
$t_1 \leftarrow$ 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

Liveness Analysis (3-address code)

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	b, e
$t_1 \leftarrow$ 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

Liveness Analysis (3-address code)

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	b, e
$t_1 \leftarrow$ 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

Liveness Analysis (3-address code)

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	b, t_0
$t_2 \leftarrow b * t_1$	t_2	b, t_1	b, t_1
ret t_2		t_2	t_2

Liveness Analysis (3-address code)

Rules for Low-Level 3-Address Code

$$\frac{\begin{array}{l} l : \mathsf{call}\ f \\ \mathsf{caller\text{-}save}(r) \end{array}}{\mathsf{def}(l, r)}\ J'_8 \qquad \frac{\begin{array}{l} l : \mathsf{ret}\ s \\ \mathsf{callee\text{-}save}(r) \end{array}}{\mathsf{use}(l, r)}\ J'_2$$

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			

Liveness

Take into account pre-colored 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			

Liveness

Take into account pre-colored 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			

Liveness

Take into account pre-colored 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			
$t_1 \leftarrow res_0$			
$t_2 \leftarrow b * t_1$	t_2		
$res_0 \leftarrow t_2$	res_0		
ret			

Liveness

Take into account pre-colored 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_1		
$t_2 \leftarrow b * t_1$	t_2		
$res_0 \leftarrow t_2$	res_0		
ret			

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ 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			

Liveness

Take into account pre-colored 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 :			
$t_0 \leftarrow e - 1$			
$arg_2 \leftarrow t_0$			
$arg_1 \leftarrow b$	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			

Liveness

Take into account pre-colored 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_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			

Liveness

Take into account pre-colored 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 :			
$t_0 \leftarrow e - 1$	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			

Liveness

Take into account pre-colored 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 :			
$t_0 \leftarrow e - 1$	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			

Liveness

Take into account pre-colored 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 :			
$t_0 \leftarrow e - 1$	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			

Liveness

Take into account pre-colored registers

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	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			

Liveness

Take into account pre-colored registers

program	def	use	live-in
pow :			
$b \leftarrow arg_1$			
$e \leftarrow arg_2$			
if $(e == 0)$ then done else recurse			
done :			
$res_0 \leftarrow 1$	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			

Liveness

Take into account pre-colored registers

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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ 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			

Liveness

Take into account pre-colored registers

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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ 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			

Liveness

Take into account pre-colored 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		
$arg_2 \leftarrow t_0$	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			

Liveness

Take into account pre-colored 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		
$arg_2 \leftarrow t_0$	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	

Liveness

Take into account pre-colored 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		
$arg_2 \leftarrow t_0$	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	

Liveness

Take into account pre-colored 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		
$arg_2 \leftarrow t_0$	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	

Liveness

Take into account pre-colored 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		
$arg_2 \leftarrow t_0$	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	

Liveness

Take into account pre-colored 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		
$arg_2 \leftarrow t_0$	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	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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		
$arg_2 \leftarrow t_0$	arg_2		
$arg_1 \leftarrow b$	arg_1	b	
call pow	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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		
$arg_2 \leftarrow t_0$	arg_2	t_0	
$arg_1 \leftarrow b$	arg_1	b	
call pow	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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		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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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		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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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		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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored registers

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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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	

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	b, arg_2
call pow	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	b, t_0
$arg_1 \leftarrow b$	arg_1	b	b, arg_2
call pow	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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 :			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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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			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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored 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	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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored registers

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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored registers

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	$res_0, arg_1, arg_2,$ $arg_3, arg_4, arg_5,$ arg_6, ler_7, ler_8	arg_1, arg_2	b, arg_1, arg_2
$t_1 \leftarrow res_0$	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

Liveness

Take into account pre-colored registers

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

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 :	
$res_0 \leftarrow 1$	
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 lee_9	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 :	
$res_0 \leftarrow 1$	
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 lee_9	res_0
ret	lee_9, res_0

Register Allocation:

We order the colors (machine registers) as

$$res_0, arg_1, \ldots, arg_6, ler_7, ler_8, lee_9$$

From this we construct the assignment

$$\begin{array}{lcl} b & \mapsto & lee_9 \\ e & \mapsto & res_0 \\ t_0 & \mapsto & res_0 \\ t_1 & \mapsto & res_0 \\ t_2 & \mapsto & res_0 \end{array}$$

Save Callee-Saved Regs and epilog

pow :
push lee_9
$lee_9 \leftarrow arg_1$
$res_0 \leftarrow arg_2$
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 lee9
    lee9 ← arg1
    res0 ← arg2
    if (res0 == 0) then done else recurse
done :
    res0 ← 1
    goto exitpow
recurse :
    res0 ← res0 − 1
    arg2 ← res0
    arg1 ← lee9                    (redundant)
    call pow
    res0 ← res0                    (redundant)
    res0 ← lee9 ∗ res0
    res0 ← res0                    (redundant)
    goto exitpow
exitpow :
    pop lee9
    ret
```

Optimization and Code Generation

pow :
  push lee_9
  $lee_9 \leftarrow arg_1$
  $res_0 \leftarrow arg_2$
  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
  $res_0 \leftarrow res_0$ (redundant)
  $res_0 \leftarrow lee_9 * res_0$
  $res_0 \leftarrow res_0$ (redundant)
  goto exitpow
exitpow :
  pop lee_9
  ret

```
pow:    pushq   %rbx
        movl    %edi, %ebx
        movl    %esi, %eax
        cmpl    $0, %eax
        jne L1
        movl    $1, %eax
        goto L2
L1:     subl    $1, %eax
        movl    %eax, %esi
        call    pow
        imull   %ebx, %eax
L2:     popq %rbx
        ret
```

Optimization and Code Generation