Termination Examples

Pete Manolios
Northeastern

Formal Methods, Lecture 2

September 2008

Termination examples

- Does the following terminate?

```
(app x y) =
(if (endp x)
    y
   (cons (car x) (app (cdr x) y)))
```

- Yes
- Recurring down a list

Termination examples

- Does the following terminate?

```
(foo x y) =
(if (consp x)
    y
   (cons (car x) (foo (cdr x) y)))
```

- No
- Consider (foo nil nil)

Termination examples

- Does the following terminate?

```
(foo x y) =
(if (consp x)
    y
   (cons (car x) (app (cdr x) y)))
```

- Yes
- Not recursive

Termination examples

- Does the following terminate?

```
(f x) =
(if (endp x)
    1
  (+ (f (car x)) (f (cdr x))))
```

- Yes
- Recurring down a tree

Termination examples

- Does the following terminate?

  (f x) =

  (not (f x))
- No
- Does it lead to unsoundness?
- Yes
- Ouch! We can now prove *any* theorem.

Termination examples

- Does the following terminate?

```
(h n) =
(if (= n 0)
    nil
  (h (- n 1)))
```

- No
- Does introducing g lead to unsoundness?
- No

Termination examples

- Does the following terminate?

```
(g n) =
(if (= n 0)
    nil
   (cons nil (g (- n 1))))
```

- No
- Does introducing g lead to unsoundness?
- Yes

Termination examples

- Does the following terminate?

```
(f n) =
(cond ((or (zp n) (<= n 1))
       n)
      ((evenp n)
       (f (/ n 2)))
      (t (f (1+ n))))
```

- Yes
- Why?

Termination examples

- Does the following terminate?

```
(f n) =
(cond ((or (zp n) (<= n 1))
       n)
      ((evenp n)
       (f (/ n 2)))
      (t (f (1+ (* 2 n)))))
```

- No: 3, 7, 15, ...
- Does introducing f lead to unsoundness?
- Definitely not

Termination examples

- Does the following terminate?

```
(c n) =
(cond ((or (zp n) (<= n 1))
       n)
      ((evenp n)
       (c (/ n 2)))
      (t (c (1+ (* 3 n)))))
```

- Probably
- Does introducing c lead to unsoundness?
- Definitely not

Termination examples

- Does the following terminate?

```
(ack x y) =
(cond ((zp x)
         (1+ y))
       ((zp y)
        (ack (1- x) 1))
       (t (ack (1- x) (ack x (1- y)))))
```

- Yes
- Why?
- Challenge problem: What is the largest n for which compute (ack n n)?
- Let's try it w/ ACL2s. See the attached ACL2s file.