Rewrite Systems

3. Orderings

HW#1 (Scheme)

```
((sort nil) . nil)
((sort (cons x y)) . (add2list x (sort y)))
((max 0 x). x)
((max x 0). x)
((max (s x) (s y)) . (s (max x y)))
((min 0 x). 0)
((min x 0). 0)
((min (s x) (s y)) . (s (min x y)))
((add2list x nil) . (cons x nil))
((add2list x (cons y z)) .
                    (cons (min x y) (add2list (max x y) z) ))
```

2

HW#1 (Human)

sort(nil) → nil
sort(cons(x,y)) → add2list(x,sort(y))
max(0,x) → x
max(x,0) → x
max(s(x),s(y)) → s(max(x,y))
min(0,x) → 0
min(x,0) → 0
min(s(x),s(y)) → s(min(x,y))
add2list(x,nil) → cons(x,nil)
add2list(x,cons(y,z)) → cons(min(x,y),add2list(max(x,y),z))

3

HW#2

Prove termination (or nontermination) of your solution to hw#1.

4

Alternative

$\text{min}(x,y) \to \text{other}(\text{max}(x,y),x,y)$

$\text{other}(x,x,y) \to y$

$\text{other}(y,x,y) \to x$

5

Termination Method

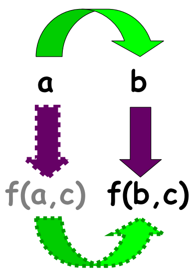
A rewrite system terminates

iff

its rules decrease
for some reduction (well-founded monotonic) ordering

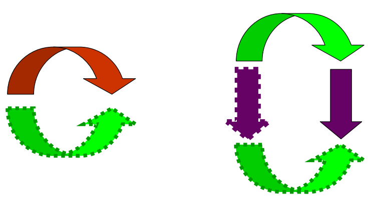
6

Monotonicity



7

Standard Method



8

Monotonic Interpretation

$$f(f(x)) \quad \longrightarrow \quad f(g(f(x)))$$

[f](a,n) = (1,n+a)
[g](a,n) = (0,n)
[const] = (0,0)

(x,y) > (x′,y′) iff x=x′,y>y′

[ff](0,n) = (1,n+1) > (1,n) = [fgf](0,n)
[ff](1,n) = (1,n+2) > (1,n+1) = [fgf](1,n)

9

Alternate Interpretation

$$f(f(x)) \quad \longrightarrow \quad f(g(f(x)))$$

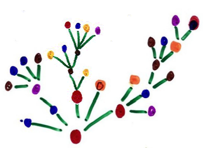
[*f*](m,n) = (n+m,m)
[*g*](m,n) = (n,m)
[const] = (1,1)

[ff](m,n) = (n+2m,n+m) >
(n+2m,m) = [fgf](m,n)

10

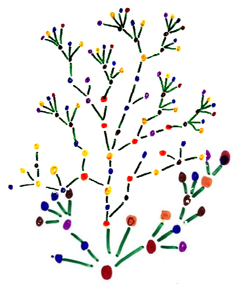
Path Orderings

Gremlins



12

Gremlins



13

Path orderings

- Recursive path ordering
- Lexicographic path ordering
- Semantic path ordering
- General path ordering
- Monotonic path ordering

14

$$
\begin{aligned}
Dt &\rightarrow 1 \\
D(\text{constant}) &\rightarrow 0 \\
D(x+y) &\rightarrow Dx + Dy \\
D(x \times y) &\rightarrow (y \times Dx) + (x \times Dy) \\
D(x - y) &\rightarrow Dx - Dy\,.
\end{aligned}
$$

15

$$(x \cdot y) \cdot z \quad \rightarrow \quad x \cdot (y \cdot z)$$

$$\langle \text{size, first multiplicand, second multiplicand} \rangle$$

16

Try Recursion

$$\begin{array}{lcl} f(g(x)) & \rightarrow & g(g(f(x))) \\ f(g(x)) & \rightarrow & g(g(g(x))) \end{array}$$

$$\langle \text{top, argument} \rangle.$$

$f > g$

17

Bound

$$\begin{array}{lcl} f(g(x)) & \rightarrow & g(g(f(x))) \\ f(g(x)) & \rightarrow & g(g(g(x))) \end{array}$$

$$\langle \text{top, argument} \rangle.$$

$$f(x), g(x) \succ x.$$

18

Lexicographic Path Ordering

$$\frac{s_i \succeq_{\text{lpo}} t}{f(s_1,\ldots,s_m) \succ_{\text{lpo}} t}$$

$$\frac{f \succ g \quad f(s_1,\ldots,s_m) \succ_{\text{lpo}} t_1,\ldots,t_n}{f(s_1,\ldots,s_m) \succ_{\text{lpo}} g(t_1,\ldots,t_n)} \qquad \frac{f \approx g \quad \langle s_1,\ldots,s_m\rangle \approx_{\text{lex}} \langle t_1,\ldots,t_n\rangle}{f(s_1,\ldots,s_m) \approx_{\text{lpo}} g(t_1,\ldots,t_n)}$$

$$\frac{f \approx g \quad \langle s_1,\ldots,s_m\rangle \succ_{\text{lex}} \langle t_1,\ldots,t_n\rangle \quad f(s_1,\ldots,s_m) \succ_{\text{lpo}} t_2,\ldots,t_n}{f(s_1,\ldots,s_m) \succ_{\text{lpo}} g(t_1,\ldots,t_n)}$$

19

$$\begin{array}{rclrcl}
1\cdot x & \to & x & x\cdot 1 & \to & x \\
x^-\cdot x & \to & 1 & x\cdot x^- & \to & 1 \\
1^- & \to & 1 & x^{--} & \to & x \\
y^-\cdot(y\cdot z) & \to & z & y\cdot(y^-\cdot z) & \to & z \\
(x\cdot y)\cdot z & \to & x\cdot(y\cdot z) & (x\cdot y)^- & \to & y^-\cdot x^-
\end{array}$$

20

$$\begin{array}{rcl}
ack(0,y) & \to & succ(y) \\
ack(succ(x),0) & \to & ack(x,succ(0)) \\
ack(succ(x),succ(y)) & \to & ack(x,ack(succ(x),y)) \ .
\end{array}$$

21

Multiset Path Ordering

$$\frac{s_i \succsim_{\text{mpo}} t}{f(s_1,\dots,s_m) \succ_{\text{mpo}} t} \qquad \frac{f \succ g \quad f(s_1,\dots,s_m) \succ_{\text{mpo}} t_1,\dots,t_n}{f(s_1,\dots,s_m) \succ_{\text{mpo}} g(t_1,\dots,t_n)}$$

$$\frac{f \approx g \quad [s_1,\dots,s_m] \approx_{\text{bag}} [t_1,\dots,t_n]}{f(s_1,\dots,s_m) \approx_{\text{mpo}} g(t_1,\dots,t_n)} \qquad \frac{f \approx g \quad [s_1,\dots,s_m] \succ_{\text{bag}} [t_1,\dots,t_n]}{f(s_1,\dots,s_m) \succ_{\text{mpo}} g(t_1,\dots,t_n)}$$

22

$$\begin{aligned}
\neg\neg x &\rightarrow x \\
\neg(x \vee y) &\rightarrow (\neg x) \wedge (\neg y) \\
\neg(x \wedge y) &\rightarrow (\neg x) \vee (\neg y) \\
x \wedge (y \vee z) &\rightarrow (x \wedge y) \vee (x \wedge z) \\
(y \vee z) \wedge x &\rightarrow (x \wedge y) \vee (x \wedge z)
\end{aligned}$$

23

Ramsey's Theorem

Infinite complete graph
Finitely colored edges

Monochrome infinite clique

24

Simple Case

- Two colors: yes and no
  - Colorblind reduction
- Color yes is transitive

25

Simple Case

- Extend yes as long as possible
- If can forever, then done (all yes)
- If not, then repeat
- If repeats forever, then done (all no)

26

$S := V$
$R := \emptyset$
do forever
    $x :\in S$
    $R := R \cup \{x\}$
    $S := S \setminus \{x\}$
    $W := \{s \in S \mid c(x, s) = \text{white}\}$

$$S := \begin{cases} W & \textbf{if } |W| = \infty \\ S \setminus W & \textbf{otherwise} \end{cases}$$

$W := \{x \in R \mid \forall y \in R.\ y \neq x \rightarrow c(x, y) = \text{white}\}$

$$\textbf{return} \begin{cases} W & \textbf{if } |W| = \infty \\ R \setminus W & \textbf{otherwise} \end{cases}$$

27

Ramsey's Theorem

Infinite complete multi-graph
Finitely colored multi-edges

Monochrome infinite clique

can have multiple multi-edges

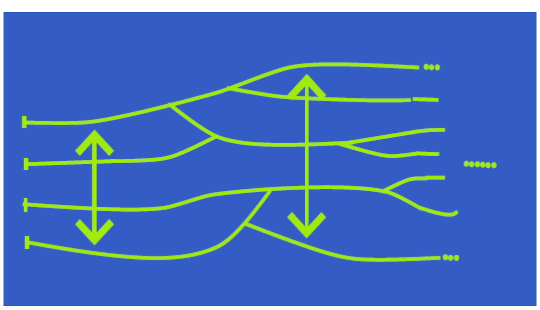
28

Well-Quasi-Orderings

- Quasi-ordering
  - Reflexive
  - Transitive
- Well-quasi-ordered
  - No infinite decreasing chains
  - No infinite anti-chains

29

Wqo



30

Well-Quasi-Order

Definition. *A set A is Well Quasi Ordered under $\precsim$ if for all infinite sequences from A:*

$$a_1, a_2, a_3, \ldots$$

there exists some $i < j$ such that $a_i \precsim a_j$.

31

Wqos

- Every infinite sequence has an infinite non-decreasing chain
- Every extension is well-founded

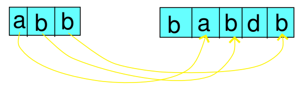
32

Good & Bad

- A sequence is bad if there is no such pair
- It is good if it has at least one pair
- A qo is a wqo if all sequences are good

33

Homeomorphic Embedding



34

Higman's Lemma

- Every infinite sequence of words (over a finite alphabet) includes an embedding.

35

Minimal Bad Sequence

- acd eef afda …
- afda ab acd …
- ab eef afda …
- ab acd eef afda …
- ab afda acd …
- …

36

Minimal Bad Sequence

- acd eef afda ...
- afda ab acd ...
- ab eef afda ...
- ab acd eef afda ...
- ab afda acd ...
- ...

37

Minimal Bad Sequence

- ab eef afda ...
- ab acd eef afda ...
- ab afda acd ...
- ...

38

Minimal Bad Sequence

- ab eef afda ...
- ab acd eef afda ...
- ab afda acd ...
- ...

39

Minimal Bad Sequence

- ab acd eef afda ...

40

Minimal Bad Sequence

- ab acd eef afda ...

41

Minimal Bad Sequence

- ab acd afda ...

42

Contradiction

- ab acd afda ... aacafad ...

43