Compilers and Formal Languages

Email:	christian.urban at kcl.ac.uk
Office:	S1.27 (1st floor Strand Building)
Slides:	KEATS (also home work is there)

2nd CW

Remember we showed that

$$der\ c\ (r^+) = (der\ c\ r) \cdot r^*$$

2nd CW

Remember we showed that

$$der\ c\ (r^{+}) = (der\ c\ r) \cdot r^{*}$$

Does the same hold for $r^{\{n\}}$ with $n > 0$

$$der\ c\ (r^{\{n\}}) = (der\ c\ r) \cdot r^{\{n-1\}} \ ?$$

2nd CW

- *der*

$$der\ c\ (r^{\{n\}}) \stackrel{\text{def}}{=} \begin{cases} \varnothing & \text{if } n = 0 \\ der\ c\ (r \cdot r^{\{n-1\}}) & \text{o'wise} \end{cases}$$

- *mkeps*

$$mkeps(r^{\{n\}}) \stackrel{\text{def}}{=} [\underbrace{mkeps(r), \ldots, mkeps(r)}_{n\ times}]$$

- *inj*

$$\begin{array}{lcl} inj\ r^{\{n\}}\ c\ (v_1, [vs]) & \stackrel{\text{def}}{=} & [inj\ r\ c\ v_1 :: vs] \\ inj\ r^{\{n\}}\ c\ Left(v_1, [vs]) & \stackrel{\text{def}}{=} & [inj\ r\ c\ v_1 :: vs] \\ inj\ r^{\{n\}}\ c\ Right([v :: vs]) & \stackrel{\text{def}}{=} & [mkeps(r) :: inj\ r\ c\ v :: vs] \end{array}$$

Compilers in Boeings 777

They want to achieve triple redundancy in hardware faults.

They compile 1 Ada program to

- Intel 80486
- Motorola 68040 (old Macintosh's)
- AMD 29050 (RISC chips used often in laser printers)

Proofs about Rexps

Remember their inductive definition:

$$
\begin{array}{rcl}
r & ::= & \varnothing \\
  & \mid & \epsilon \\
  & \mid & c \\
  & \mid & r_1 \cdot r_2 \\
  & \mid & r_1 + r_2 \\
  & \mid & r^*
\end{array}
$$

If we want to prove something, say a property $P(r)$, for all regular expressions r then ...

Proofs about Rexp (2)

- P holds for $\varnothing$, ϵ and $\mathbf{c}$
- P holds for $r_1 + r_2$ under the assumption that P already holds for r_1 and r_2.
- P holds for $r_1 \cdot r_2$ under the assumption that P already holds for r_1 and r_2.
- P holds for r^* under the assumption that P already holds for r.

$$
\begin{array}{lcl}
zeroable(\varnothing) & \stackrel{\text{def}}{=} & true\\
zeroable(\epsilon) & \stackrel{\text{def}}{=} & false\\
zeroable(c) & \stackrel{\text{def}}{=} & false\\
zeroable(r_1 + r_2) & \stackrel{\text{def}}{=} & zeroable(r_1) \wedge zeroable(r_2)\\
zeroable(r_1 \cdot r_2) & \stackrel{\text{def}}{=} & zeroable(r_1) \vee zeroable(r_2)\\
zeroable(r^*) & \stackrel{\text{def}}{=} & false
\end{array}
$$

$zeroable(r)$ if and only if $L(r) = \{\}$

Correctness of the Matcher

- We want to prove

$$\mathit{matches}\ r\ s \text{ if and only if } s \in L(r)$$

where $\mathit{matches}\ r\ s \stackrel{\text{def}}{=} \mathit{nullable}(\mathit{ders}\ s\ r)$

Correctness of the Matcher

- We want to prove

$$matches\ r\ s \text{ if and only if } s \in L(r)$$

where $matches\ r\ s \stackrel{\text{def}}{=} nullable(ders\ s\ r)$

- We can do this, if we know

$$L(der\ c\ r) = Der\ c\ (L(r))$$

Induction over Strings

- case $[]$:

  We need to prove

$$\forall r.\ \ \textit{nullable}(\textit{ders}\ []\ r)\ \ \Leftrightarrow\ \ []\in L(r)$$

$$\textit{nullable}(\textit{ders}\ []\ r)\ \stackrel{\text{def}}{=}\ \textit{nullable}\ r\ \ \Leftrightarrow\ \ldots$$

Induction over Strings

- case $c :: s$

  We need to prove

  $$\forall r.\;\; \mathit{nullable}(\mathit{ders}\ (c :: s)\ r) \;\Leftrightarrow\; (c :: s) \in L(r)$$

  We have by IH

  $$\forall r.\;\; \mathit{nullable}(\mathit{ders}\ s\ r) \;\Leftrightarrow\; s \in L(r)$$

  $$\mathit{ders}\ (c :: s)\ r \stackrel{\text{def}}{=} \mathit{ders}\ s\ (\mathit{der}\ c\ r)$$

Induction over Regexps

- The proof hinges on the fact that we can prove

$$L(der\ c\ r) = Der\ c\ (L(r))$$

Some Lemmas

- $Der\ c\ (A \cup B) = (Der\ c\ A) \cup (Der\ c\ B)$

- If $[] \in A$ then
  $$Der\ c\ (A\,@\,B) = (Der\ c\ A)\,@\,B \;\cup\; (Der\ c\ B)$$

- If $[] \notin A$ then
  $$Der\ c\ (A\,@\,B) = (Der\ c\ A)\,@\,B$$

- $Der\ c\ (A^*) = (Der\ c\ A)\,@\,A^*$

  (interesting case)

Why?

Why does $Der\ c\ (A^*) = (Der\ c\ A) @ A^*$ hold?

$$\begin{array}{rcl}
Der\ c\ (A^*) & = & Der\ c\ (A^* - \{[]\}) \\
& = & Der\ c\ ((A - \{[]\}) @ A^*) \\
& = & (Der\ c\ (A - \{[]\})) @ A^* \\
& = & (Der\ c\ A) @ A^*
\end{array}$$

using the facts $Der\ c\ A = Der\ c\ (A - \{[]\})$ and
$(A - \{[]\}) @ A^* = A^* - \{[]\}$