Shared Forest Algorithms

CS674

Spring 2007

Shared Forest

- Mapped grammar construction
- Circuit construction
- Compute things we are interested in in the shared forest data structure
- Today's algorithms assume the shared forest in non-recursive
- Algorithms iterate in bottom up (shorter spans first) or top-down (longer spans first) order

Inside Tree Set

- Where A→W is a lexical entry of the shared forest grammar, I(A→W) = …
- Where A→B C is a production of the shared forest grammar, I(A→B C) = …
- Where A→α is a production of the shared forest grammar, I(A→α) = …
- f is the function from shared forest labels to syntactic labels

Rule of Product

To do a, you have to do b and then do c.

There are x ways of doing b.

There are y ways of doing c.

Then there are xy ways of doing a.

Shared forest rule A→B C

Construct a tree t with …

- t(ε) = f(A) and rule(t,ε) is binary one way
- subtree(t,0) ∈ I(B)
  cardinality of I(B) ways
- subtree(t,1) ∈ I(C)
  cardinality of I(C) ways

1 * |I(B)| * |I(C)| ways of doing it, by rule of product

Rule of Sum

- To do a, you have to do b or do c.
- Ways of doing b and c are disjoint.
- There are x ways of doing b, and y ways of doing c.
- Then there are x+y ways of doing a.

Rule of Sum

PF-INSIDE-TREE-COUNT(N, Σ, P, R, L)

```
1  Initialize integer array i[N ∪ L ∪ P]
2  for x in L
3      do i[x] ← 1
4  for p in P in bottom-up order
5      do i[p] ← ∏_{y in rhs(p)} i[y]
6          i[lhs(p)] ← i[lhs(p)] + i[p]
7  return i
```

Probabilistic Inside Algorithm

At each product step, multiply by
$p(f(A) \rightarrow f(\alpha_1) \ldots f(\alpha_n))$

At base, initialize with $p(A \rightarrow w)$

Probabilisitic Inside Max

Use the max operation as the sum

PF-Complete-Tree-Count(N, Σ, P, R, L)

1 $i[N \cup P \cup L] \leftarrow$ PF-Inside-Tree-Count(N, Σ, P, R, L)
2 Initialize $c[N \cup P \cup L] \leftarrow 0$
3 **for** x in R
4 **do** $c[x] \leftarrow i[x]$
5 **for** p in P in top-down order
6 **do** $c[p] \leftarrow \frac{i[p]}{i[lhs(p)]} c[lhs(p)]$
7 **for** y in $rhs(p)$
8 **do** $c[y] \leftarrow c[y] + c[p]$
9 **for** p in L
10 **do** $c[p] \leftarrow \frac{i[p]}{i[lhs(p)]} c[lhs(p)]$
11 return c