POPL2 (09-04-2020)

Srijith P.K.

Sub-goal order

- Choice in premises
- Multiplication : times (m, n, p)

$$\begin{aligned} 0 \times n &= 0 \\ (m+1) \times n &= (m \times n) + n \end{aligned}$$

Sub-goal order

- Choice in premises
- Multiplication : times (m,n,p)

$$\begin{aligned} 0 \times n &= 0 \\ (m+1) \times n &= (m \times n) + n \end{aligned}$$

$$\frac{}{\mathsf{times}(\mathsf{z}, N, \mathsf{z})}\ \mathsf{tz} \qquad \frac{\mathsf{times}(M, N, P) \quad \mathsf{plus}(P, N, Q)}{\mathsf{times}(\mathsf{s}(M), N, Q)}\ \mathsf{ts}$$

Multiplication

- Compute 1 * 2 = Q

$$\frac{}{\mathsf{times}(\mathsf{z}, N, \mathsf{z})}\ \mathsf{tz} \qquad \frac{\mathsf{times}(M, N, P) \quad \mathsf{plus}(P, N, Q)}{\mathsf{times}(\mathsf{s}(M), N, Q)}\ \mathsf{ts}$$

Multiplication

- Compute 1 * 2 = Q

$$\dfrac{\dfrac{}{\mathsf{times}(\mathsf{z}, N, \mathsf{z})}\,\mathsf{tz} \qquad \dfrac{\mathsf{times}(M, N, P) \quad \mathsf{plus}(P, N, Q)}{\mathsf{times}(\mathsf{s}(M), N, Q)}\,\mathsf{ts}}{\vdots \qquad\qquad \vdots}$$
$$\dfrac{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P) \quad \mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}$$

Multiplication

- Compute 1 * 2 = Q

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{times}(\mathsf{z}, N, \mathsf{z})}\,\mathsf{tz} \qquad \dfrac{\mathsf{times}(M, N, P) \quad \mathsf{plus}(P, N, Q)}{\mathsf{times}(\mathsf{s}(M), N, Q)}\,\mathsf{ts}}{\vdots}}{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P) \quad \mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}$$

$$\dfrac{\dfrac{\dfrac{}{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P)}\,\mathsf{tz}\,(P = \mathsf{z}) \qquad \vdots}{\mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}$$

Multiplication

- Compute 1 * 2 = Q

$$\dfrac{\dfrac{}{\mathsf{times}(\mathsf{z}, N, \mathsf{z})}\,\mathsf{tz} \qquad \dfrac{\mathsf{times}(M, N, P) \quad \mathsf{plus}(P, N, Q)}{\mathsf{times}(\mathsf{s}(M), N, Q)}\,\mathsf{ts}}{\dfrac{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P) \quad \mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}}$$

$$\dfrac{\dfrac{}{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P)}\,\mathsf{ts}\ (P = \mathsf{z}) \qquad \dfrac{}{\mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{pz}\ (Q = \mathsf{s}(\mathsf{s}(\mathsf{z})))}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}.$$

Multiplication

- Compute 1 * 2 = Q

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{times}(\mathsf{z}, N, \mathsf{z})}\,\mathsf{tz} \qquad \dfrac{\mathsf{times}(M, N, P) \quad \mathsf{plus}(P, N, Q)}{\mathsf{times}(\mathsf{s}(M), N, Q)}\,\mathsf{ts}}{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P) \quad \mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)} \qquad \dfrac{\mathsf{plus}(P_1, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q_1)}{\mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ps}\ (P = \mathsf{s}(P_1), Q = \mathsf{s}(Q_1))}{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P) \quad \mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}.$$

Multiplication

- Compute 1 * 2 = Q

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{times}(\mathsf{z}, N, \mathsf{z})}\,\mathsf{tz} \qquad \dfrac{\mathsf{times}(M, N, P) \quad \mathsf{plus}(P, N, Q)}{\mathsf{times}(\mathsf{s}(M), N, Q)}\,\mathsf{ts}}{\vdots}}{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P) \quad \mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}}{\vdots}}{\dfrac{\mathsf{plus}(P_2, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q_2)}{\mathsf{plus}(P_1, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q_1)}\,\mathsf{ps}\ (P_1 = \mathsf{s}(P_2), Q_1 = \mathsf{s}(Q_2))}}{\vdots}}{\dfrac{\mathsf{times}(\mathsf{z}, \mathsf{s}(\mathsf{s}(\mathsf{z})), P) \quad \mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}{\mathsf{plus}(P, \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ps}\ (P = \mathsf{s}(P_1), Q = \mathsf{s}(Q_1))}}{\mathsf{times}(\mathsf{s}(\mathsf{z}), \mathsf{s}(\mathsf{s}(\mathsf{z})), Q)}\,\mathsf{ts}$$

Computation can go indefinitely

Sub-goal order

- order in which subgoals are solved can have a strong impact on the computation.
  - search either completes in two steps or does not terminate.
- attack the subgoals in left-to-right order.
- two definitions, entirely equivalent from the logical point of view, can be very different operationally

Unification

- determine if the conjecture matches the conclusion of an inference rule ⇔ goal unifies with the head of a clause
- Rules and Goals may contain variables
- plus(s(z), s(z),R) and the clause plus(s(M),N, s(P)) :- plus(M,N, P).
- match up corresponding subterms.
- M = z, N = s(z), and R = s(P).
- solve plus(z, s(z), P) = plus(z,N,N) which sets N = s(z) and P = s(z).
- This process is called *unification*, and the equations for the variables we generate represent the *unifier*.

Prolog

$$\frac{J_1 \ldots J_n}{J}\ R \qquad J \leftarrow J_1, \ldots, J_n. \qquad J \textit{ if } J_1 \textit{ and } \cdots \textit{ and } J_n.$$

left-pointing arrow is rendered as ':-'

Prolog terminology for an inference rule is a *clause*, where J is the *head* of the clause and J1, . . . , Jn is the body.

"search for a clause whose head matches the goal"

Prolog

$$\frac{}{\mathsf{even}(\mathsf{z})}\ \mathsf{evz} \qquad \frac{\mathsf{even}(N)}{\mathsf{even}(\mathsf{s}(\mathsf{s}(N)))}\ \mathsf{evs}$$

$$\frac{\mathsf{plus}(M, N, P)}{\mathsf{plus}(\mathsf{s}(M), N, \mathsf{s}(P))}\ \mathsf{ps} \qquad \frac{}{\mathsf{plus}(\mathsf{z}, N, N)}\ \mathsf{pz}.$$

$$\frac{}{\mathsf{times}(\mathsf{z}, N, \mathsf{z})}\ \mathsf{tz} \qquad \frac{\mathsf{times}(M, N, P) \quad \mathsf{plus}(P, N, Q)}{\mathsf{times}(\mathsf{s}(M), N, Q)}\ \mathsf{ts}$$

Prolog

$$\frac{}{\mathsf{even}(\mathsf{z})}\ \mathsf{evz} \qquad \frac{\mathsf{even}(N)}{\mathsf{even}(\mathsf{s}(\mathsf{s}(N)))}\ \mathsf{evs}$$

```
even(z).
even(s(s(N))) :- even(N).
```

$$\frac{\mathsf{plus}(M,N,P)}{\mathsf{plus}(\mathsf{s}(M),N,\mathsf{s}(P))}\ \mathsf{ps} \qquad \frac{}{\mathsf{plus}(\mathsf{z},N,N)}\ \mathsf{pz}.$$

```
plus(s(M), N, s(P)) :- plus(M, N, P).
plus(z, N, N).
```

$$\frac{}{\mathsf{times}(\mathsf{z},N,\mathsf{z})}\ \mathsf{tz} \qquad \frac{\mathsf{times}(M,N,P) \quad \mathsf{plus}(P,N,Q)}{\mathsf{times}(\mathsf{s}(M),N,Q)}\ \mathsf{ts}$$

```
times(z, N, z).
times(s(M), N, Q) :-
        times(M, N, P),
        plus(P, N, Q).
```

Clauses are tried in the order they are presented in the program. Subgoals are solved in the order they are presented in the body of a clause.

Type Predicates

- Lists : [a,b,c] or [a | [b | [c | []]]] or [a, b | [c, []]]
- list a, b, c would be represented as cons(a, cons(b, cons(c, nil)))
- plus(z, N, N).
- plus(s(M), N, s(P)) :- plus(M, N, P).
- plus(s(z), [a, b, c], s([a, b, c])).
- what does it mean to add 1 and a list ?
- Definition above lacks types !

Type predicates

define a predicate nat with the rules

$$\frac{}{\mathsf{nat}(\mathsf{z})}\ \mathsf{nz} \qquad \frac{\mathsf{nat}(N)}{\mathsf{nat}(\mathsf{s}(N))}\ \mathsf{ns}$$

$$\frac{\mathsf{nat}(N)}{\mathsf{plus}(\mathsf{z}, N, N)}\ \mathsf{pz} \qquad \frac{\mathsf{plus}(M, N, P)}{\mathsf{plus}(\mathsf{s}(M), N, \mathsf{s}(P))}\ \mathsf{ps}$$

```
nat(z).
nat(s(N)) :- nat(N).

plus(z, N, N) :- nat(N).
plus(s(M), N, s(P)) :- plus(M, N, P).
```

List Types

```
list([]).
list([X|Xs]) :- list(Xs).


natlist([]).
natlist([N|Ns]) :- nat(N), natlist(Ns).
```

List operations

$$\frac{}{\mathsf{member}(X, \mathsf{cons}(X, Ys))}$$

$$\frac{\mathsf{member}(X, Ys)}{\mathsf{member}(X, \mathsf{cons}(Y, Ys))}$$

```
member(X, [X|Ys]).
member(X, [Y|Ys]) :- member(X, Ys).
```

```
?- member(a, [a,b,a,c]).
```

```
?- member(X, [a,b,a,c]).
```

List operations

- Member predicate is inefficient : when we find the first matching element there is no need to traverse the remainder of the list, although the member predicate above will always do so.
- to only check membership, or find the first occurrence of an element in a list

List operations

- Member predicate is inefficient : when we find the first matching element there is no need to traverse the remainder of the list, although the member predicate above will always do so.
- to only check membership, or find the first occurrence of an element in a list

$$\frac{}{\mathsf{member}(X, \mathsf{cons}(X, Ys))} \qquad \frac{X \neq Y \quad \mathsf{member}(X, Ys)}{\mathsf{member}(X, \mathsf{cons}(Y, Ys))}$$

```
member1(X, [X|Ys]).
member1(X, [Y|Ys]) :- X \= Y, member1(X, Ys).
```

```
?- member1(a, [a,b,a,c]).
```

```
?- member1(X, [a,b,a,c]).
```

List operations

```
prefix([], Ys).
prefix([X|Xs], [X|Ys]) :- prefix(Xs, Ys).

?- prefix(Xs,[a,b,c,d]).
```

List operations

```
prefix([], Ys).
prefix([X|Xs], [X|Ys]) :- prefix(Xs, Ys).
```

```
?- prefix(Xs,[a,b,c,d]).
Xs = [];
Xs = [a];
Xs = [a,b];
Xs = [a,b,c];
Xs = [a,b,c,d].
```

```
append([], Ys, Ys).
append([X|Xs], Ys, [X|Zs]) :- append(Xs, Ys, Zs).
```

```
prefix2(Xs, Ys) :- append(Xs, _, Ys).
```