

Note: You should not use D or E , as they already have built in meanings....
 That said, I won't fix it in these notes. Instead, I'll just clear the built-in meanings.
 (This is really bad practice!)

```
Unprotect[D, E];
ClearAll[D, E];
```

Halfway through, you introduce $E[z]$ and $D[z]$ I assume they're the same as E and D .

```
E[z] = E;
D[z] = D;
```

Note : D^* , E^* , D and E are square matrices ($n \times n$);
 X is column vector ($n \times 1$);
 $H1^*[r]$, $H2^*[r]$, $H1[r]$ and $H2[r]$ are also column vectors of ($n \times 1$);

Here's a recursive definition:

```
sub[1] = D*.X + H1*[1];

reps = {D* -> D, H1*[n_] -> H1[n], E* -> E, H2*[n_] -> H2[n]};

sub[r_Integer? (4 > # > 1 || # >= 14 &)] :=
  (*sub[r]=*)D*. (Distribute /@ (sub[r - 1] /. reps)) + H1*[r]

sub[r_Integer? (14 > # >= 4 &)] := (*sub[r]=*)E*. (Distribute /@ (sub[r - 1] /. reps)) + H2*[r]
```

Since `Dot[]` will not expand symbolic arguments over the $+$'s, I introduced `Distribute`.

The commented out code implements Memoization. See
 Functions That Remember Values They Have Found.

Note that I had to break the recursive definition into 2 parts. One for D and $H1$ and the other for E and $H2$.

■ I've changed all of your "=" into "==" in order to check my above definitions

```
sub[1] == D*.X + H1*[1]
```

True

```
sub[2] == D*. (D.X + H1[1]) + H1*[2]
```

True

Note : the above `sub[2]` is derived from

```
sub[2] = D*. (sub[1] /. {D* -> D, H1*[1] -> H1[1]}) + H1*[2]
```

```
sub[3] == D*. (D.D.X + D.H1[1] + H1[2]) + H1*[3]
```

True

Note : the above `sub[3]` is derived from

```
sub[3] = D*. (sub[2] /. {D* -> D, H1*[2] -> H1[2]}) + H1*[3]
```

```
sub[4] == E*. (D.D.D.X + D.D.H1[1] + D.H1[2] + H1[3]) + H2*[4]
```

True

Note : the above `sub[4]` is derived from

```
sub[4] = E*. (sub[3] /. {D* -> D, H1*[3] -> H1[3]}) + H2*[4]
```

```
sub[5] == E*. (E.D.D.D.X + E.D.D.H1[1] + E.D.H1[2] + E.H1[3] + H2[4]) + H2*[5]
```

True

Note : the above `sub[5]` is derived from

```
sub[5] = E*. (sub[4] /. {E* -> E, H2*[4] -> H2[4]}) + H2*[5]
```

```
sub[6] ==
E*.(E.E.D.D.D.X + E.E.D.D.H1[1] + E.E.D.H1[2] + E.E.H1[3] + E.H2[4] + H2[5]) + H2*[6]
True
```

Note : the above sub[6] is derived from

$$\text{sub}[6] = E^* \cdot (\text{sub}[5] \text{ /. } \{E^* \rightarrow E, H2^*[5] \rightarrow H2[5]\}) + H2^*[6]$$

```
sub[7] == E*.(E.E.E.D.D.D.X + E.E.E.D.D.H1[1] +
E.E.E.D.H1[2] + E.E.E.H1[3] + E.E.H2[4] + E[z].H2[5] + H2[6]) + H2*[7]
True
```

Note : the above sub[7] is derived from

$$\text{sub}[7] = E^* \cdot (\text{sub}[6] \text{ /. } \{E^* \rightarrow E, H2^*[6] \rightarrow H2[6]\}) + H2^*[7]$$

```
sub[8] == E*.(E.E.E.E.D.D.D.X + E.E.E.E.D.D.H1[1] + E.E.E.E.D.H1[2] +
E.E.E.E.H1[3] + E.E.E.H2[4] + E.E.H2[5] + E.H2[6] + H2[7]) + H2*[8]
True
```

Note : the above sub[8] is derived from

$$\text{sub}[8] = E^* \cdot (\text{sub}[7] \text{ /. } \{E^* \rightarrow E, H2^*[7] \rightarrow H2[7]\}) + H2^*[8]$$

```
sub[9] ==
E*.(E.E.E.E.E.D.D.D.X + E.E.E.E.E.D.D.H1[1] + E.E.E.E.E.D.H1[2] + E.E.E.E.E.H1[3] +
E.E.E.E.H2[4] + E.E.E.H2[5] + E.E.H2[6] + E.H2[7] + H2[8]) + H2*[9]
True
```

Note : the above sub[9] is derived from

$$\text{sub}[9] = E^* \cdot (\text{sub}[8] \text{ /. } \{E^* \rightarrow E, H2^*[8] \rightarrow H2[8]\}) + H2^*[9]$$

```
sub[10] == E*.(E.E.E.E.E.E.D.D.D.X + E.E.E.E.E.E.D.D.H1[1] +
E.E.E.E.E.E.D.H1[2] + E.E.E.E.E.E.H1[3] + E.E.E.E.E.H2[4] +
E.E.E.E.H2[5] + E.E.E.H2[6] + E.E.H2[7] + E.H2[8] + H2[9]) + H2*[10]
True
```

Note : the above sub[10] is derived from

$$\text{sub}[10] = E^* \cdot (\text{sub}[9] \text{ /. } \{E^* \rightarrow E, H2^*[9] \rightarrow H2[9]\}) + H2^*[10]$$

```
sub[11] == E*.(E.E.E.E.E.E.E.D.D.D.X + E.E.E.E.E.E.E.D.D.H1[1] +
E.E.E.E.E.E.E.D.H1[2] + E.E.E.E.E.E.E.H1[3] + E.E.E.E.E.E.H2[4] +
E.E.E.E.E.H2[5] + E.E.E.E.H2[6] + E.E.E.H2[7] + E.E.H2[8] + E.H2[9] + H2[10]) + H2*[11]
True
```

Note : the above sub[11] is derived from

$$\text{sub}[11] = E^* \cdot (\text{sub}[10] \text{ /. } \{E^* \rightarrow E, H2^*[10] \rightarrow H2[10]\}) + H2^*[11]$$

```
sub[12] ==
E*.(E.E.E.E.E.E.E.E.D.D.D.X + E.E.E.E.E.E.E.E.D.D.H1[1] + E.E.E.E.E.E.E.E.D.H1[2] +
E.E.E.E.E.E.E.E.H1[3] + E.E.E.E.E.E.E.H2[4] + E.E.E.E.E.E.H2[5] + E.E.E.E.E.H2[6] +
E.E.E.E.H2[7] + E.E.E.H2[8] + E.E.H2[9] + E.H2[10] + H2[11]) + H2*[12]
True
```

Note : the above sub[12] is derived from

$$\text{sub}[12] = E^* \cdot (\text{sub}[11] \text{ /. } \{E^* \rightarrow E, H2^*[11] \rightarrow H2[11]\}) + H2^*[12]$$

```
sub[13] == E*.(E.E.E.E.E.E.E.E.E.D.D.D.X + E.E.E.E.E.E.E.E.E.D.D.H1[1] +
E.E.E.E.E.E.E.E.E.D.H1[2] + E.E.E.E.E.E.E.E.E.H1[3] + E.E.E.E.E.E.E.E.H2[4] +
E.E.E.E.E.E.E.H2[5] + E.E.E.E.E.E.H2[6] + E.E.E.E.E.H2[7] +
E.E.E.E.H2[8] + E.E.E.H2[9] + E.E.H2[10] + E.H2[11] + H2[12]) + H2*[13]
True
```

Note : the above sub[13] is derived from

$$\text{sub}[13] = E^* \cdot (\text{sub}[12] /. \{E^* \rightarrow E, H2^*[12] \rightarrow H2[12]\}) + H2^*[13]$$

```
sub [14] == D*.(E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].D[z].D[z].D[z].X +
E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].D[z].D[z].H1[1] +
E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].D[z].H1[2] +
E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].H1[3] + E[z].E[z].E[z].E[z].
E[z].E[z].E[z].E[z].E[z].H2[4] + E[z].E[z].E[z].E[z].E[z].E[z].E[z].E[z].H2[5] +
E[z].E[z].E[z].E[z].E[z].E[z].E[z].H2[6] + E[z].E[z].E[z].E[z].E[z].E[z].H2[7] +
E[z].E[z].E[z].E[z].E[z].H2[8] + E[z].E[z].E[z].E[z].H2[9] +
E[z].E[z].E[z].H2[10] + E[z].E[z].H2[11] + E[z].H2[12] + H2[13]) + H1*[14]
```

True

Note : the above sub[14] is derived from

$$\text{sub}[14] = D^* . (\text{sub}[13] / . \{E^* \rightarrow E, H2^*[13] \rightarrow H2[13]\}) + H1^*[14]$$

```
sub [15] ==
```

$$D^* \cdot (D.E.E.E.E.E.E.E.E.E.D.D.D.X + D.E.E.E.E.E.E.E.E.E.D.D.H1[1] + D.E.E.E.E.E.E.E.E.E.D.H1[2] + D.E.E.E.E.E.E.E.E.E.E.H1[3] + D.E.E.E.E.E.E.E.E.E.H2[4] + D.E.E.E.E.E.E.E.E.E.H2[5] + D.E.E.E.E.E.E.E.E.E.H2[6] + D.E.E.E.E.E.E.E.E.E.H2[7] + D.E.E.E.E.E.E.E.E.E.H2[8] + D.E.E.E.E.E.E.E.E.E.H2[9] + D.E.E.E.E.E.E.E.E.E.H2[10] + D.E.E.E.E.E.E.E.E.E.H2[11] + D.E.E.E.E.E.E.E.E.E.H2[12] + D.E.E.E.E.E.E.E.E.E.H2[13] + H1[14]) + H1^*[15]$$

True

Note : the above sub[15] is derived from

$$\text{sub}[15] = D^* \cdot (\text{sub}[14] \text{ /. } \{D^* \rightarrow D, H1^*[14] \rightarrow H1[14]\}) + H1^*[15]$$
$$\begin{aligned} \text{sub}[16] = & D^* \cdot (D.D.E.E.E.E.E.E.E.E.E.D.D.D.X + D.D.E.E.E.E.E.E.E.E.E.D.D.H1[1] + \\ & D.D.E.E.E.E.E.E.E.E.E.E.E.D.H1[2] + D.D.E.E.E.E.E.E.E.E.E.E.E.H1[3] + \\ & D.D.E.E.E.E.E.E.E.E.E.E.H2[4] + D.D.E.E.E.E.E.E.E.E.E.E.H2[5] + D.D.E.E.E.E.E.E.E.H2[6] + \\ & D.D.E.E.E.E.E.E.H2[7] + D.D.E.E.E.E.E.H2[8] + D.D.E.E.E.E.H2[9] + D.D.E.E.E.H2[10] + \\ & D.D.E.E.H2[11] + D.D.E.H2[12] + D.D.H2[13] + D.H1[14] + H1[15]) + H1^*[16] \end{aligned}$$

True

Note : the above sub[16] is derived from

$$\text{sub}[16] = D^* \cdot (\text{sub}[15] \ /. \ \{D^* \rightarrow D, H1^*[15] \rightarrow H1[15]\}) + H1^*[16]$$

- Question

How to create a single generic code that represent all the 16 above expressions, if possible.

for instance, if I wish to substitute $r = 1$ into this desired generic code, it will express $\text{sub}[1]$ as above, if say, $r = 16$, this coveted generic code would analyse all the matrices as per $\text{sub}[16]$ equation as above.

Note also that D and E, H1 and H2 are interchangeable with respect to r.

If[3 < r < 13, E, D] and If[3 < r < 13, H2, H1];

I am trying to create a function or code in such a way that at any value of r , it will compute the matrices using that particular expression of $\text{sub}[r]$.

Perhaps the generic expression should say something like this `sub[r_]:=`