Hakupuu rajoiteongelmalle P

- Hakupuu
  - solmut rajoiteongelmia
  - juuri P
  - parillisen syvyyden solmuilla tasan yksi perillinen
  - isäsolmu on X:n suhteen ekvivalentti lasten unionin kanssa
- nimiöintipuut, jakaminen singleton joukkoihin (labeling trees) vs. enumeraatio (bisectio), arvoalueen rajaaminen rajoitteita vyöryttämällä

Nimiöintipuiden tyypit

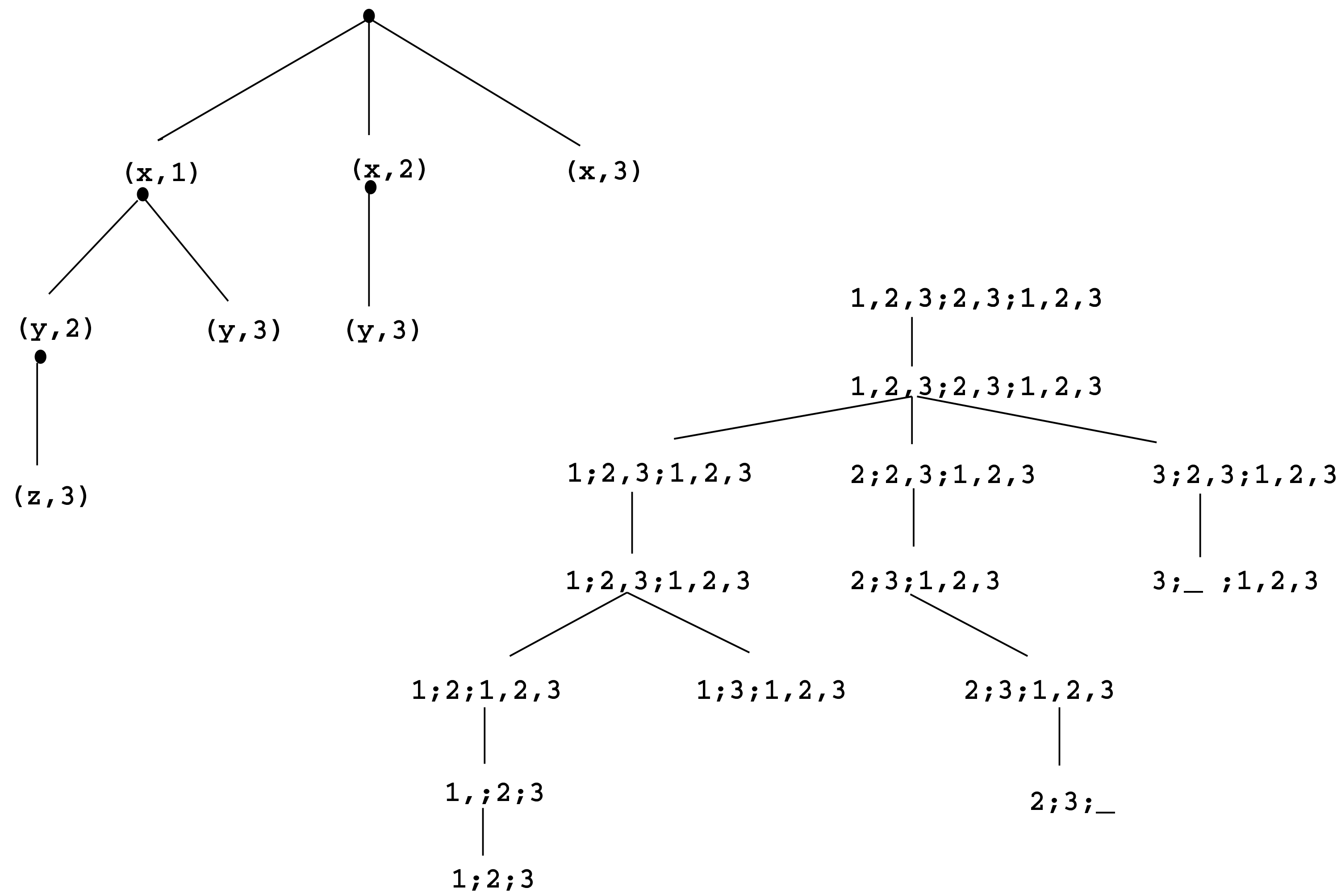
- täydellinen nimiöinti :
  - juuren lapset (x_1, d)
  - (x_j, d) , $j \in [1 \ldots n-1]$ lapset ovat muotoa (x_{j+1}, e)
  - haarat täydellisiä instantiaatioita
- karsitut nimiöintipuut: haarat kuvaavat kaikki konsistentit instantiaatiot järjestyksen $x_1 \ldots x_n$ mukaan(reduced labeling trees)
- vyörytettävät nimiöintipuut : kiinnitetty rajoitteiden vyörytysmenetelmä, solmut arvoaluelausekkeita (prop labelin trees)

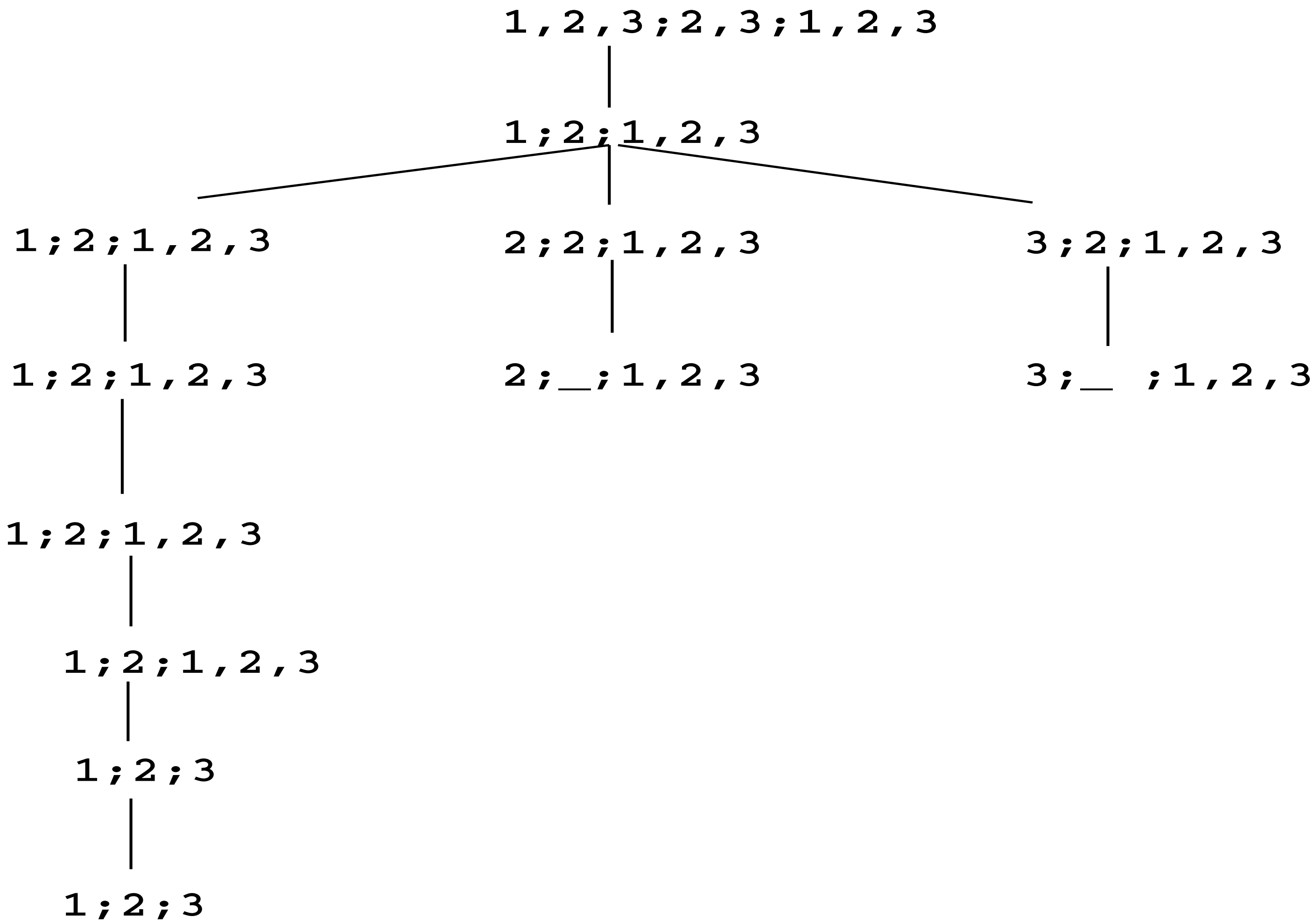
Vyörytettävät (*prop*) nimimiöintipuut

- kiinnitetty rajoitteiden vyörytysmenetelmä , (eteenpäin karsiva, suunnattukaarikonsistenssi, kaarikonsistenssi)
- solmut arvoaluelausekkeita $x_1 \in E_1, \ldots, x_n \in E_n$,
- juuri $x_1 \in D_1, \ldots, x_n \in D_n$
- solmut tasolla $2i$ muotoa $x_1 \in \{d_1\}, \ldots, x_i \in \{d_n\}, x_{i+1} \in E_{i+1}, \ldots, x_n \in E_n$
- solmut tasolla $2i + 1$ muotoa konsistenssin vyörytyksellä kustakin $2i$ tason solmusta, jos jokin $E_j = \emptyset, j \in [i + 1, \ldots, n]$ solmu on lehtisolmu, muutoin nimiöidään seuraava taso x_{i+1}:n pohjalta.

Esimerkki

- $\langle x < y, y < z; x \in \{1, 2, 3\}, y \in \{2, 3\}, z \in \{1, 23\}\rangle$
- karsiva ja vyöryttävä (eteenpäin tarkistava)
- vyöryttävä (suunnattu konsistenssi)


(x,1)
(x,2)
(x,3)
(y,2)
(y,3)
(y,3)
(z,3)
1,2,3;2,3;1,2,3
1,2,3;2,3;1,2,3
1;2,3;1,2,3
2;2,3;1,2,3
3;2,3;1,2,3
1;2,3;1,2,3
2;3;1,2,3
3;_ ;1,2,3
1;2;1,2,3
1;3;1,2,3
2;3;1,2,3
1,;2;3
2;3;_
1;2;3


1,2,3;2,3;1,2,3
1;2;1,2,3
1;2;1,2,3
2;2;1,2,3
3;2;1,2,3
1;2;1,2,3
2;_;1,2,3
3;_ ;1,2,3
1;2;1,2,3
1;2;1,2,3
1;2;3
1;2;3

5 kuningattaren ongelma

$$\langle \mathrm{C}; x_1 \in [1..5], x_2 \in [1..5], x_3 \in [1..5], x_4 \in [1..5], x_5 \in [1..5] \rangle$$

$$\begin{array}{ccccc}
\circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ \\
\star & \circ & \circ & \circ & \circ
\end{array}$$

eteenpäin karsinta

$$x_1 \in 1, x_2 \in \{3,4,5\}, x_3 \in \{2,4,5\}, x_4 \in \{2,3,5\}, x_5 \in \{2,3,4\}$$

●	○	○	○	●
●	○	○	●	○
●	○	●	○	○
●	●	○	○	○
⋆	●	●	●	●

suunnattu kaarikonsistenssi

$$x_1 \in \{1\}, x_2 \in \{3,4,5\}, x_3 \in \{2,4,5\}, x_4 \in \{2,3,5\}, x_5 \in \{2,3,4\}$$

$$\begin{array}{ccccc}
\bullet & \circ & \circ & \circ & \bullet \\
\bullet & \circ & \circ & \bullet & \circ \\
\bullet & \circ & \bullet & \triangleleft & \circ \\
\bullet & \bullet & \circ & \circ & \circ \\
\star & \bullet & \bullet & \bullet & \bullet
\end{array}$$

jokaiseen myöhempään sarakkeeseen jää ruutu

kaarikonsistenssi

$$x_1 \in 1, x_2 \in \{3,4,5\}, x_3 \in \{2,4,5\}, x_4 \in \{2,5\}, x_5 \in \{2,3,4\}$$

$$\begin{array}{ccccc}
\bullet & \circ & \circ & \circ & \bullet \\
\bullet & \circ & \triangleright & \bullet & \circ \\
\bullet & \circ & \bullet & \triangleleft & \circ \\
\bullet & \bullet & \circ & \circ & \circ \\
\star & \bullet & \bullet & \bullet & \bullet
\end{array}$$

kaikkien nimiöimättömien suhteen iteroidaan : jos tämä nimiöidään näin niin jollekin muulle ei jää nimiöitävää

Algorithm 0.1: BACKTRACKFREEPROPMODULE(C, D)

```
TYPE: domains = domain^n
TYPE: instantiation = element[n]
procedure BACKTRACKFREEPROP(j, D, success)
while D[j] ≠ ∅ and not success
  do d ← some(D[j])
     D[j] ← D[j] \ {d}
     if const(inst, j, d)
       then inst[j] ← d
            success ← (j = n)
            if not success
              then prop(j, D, failure)
                   if not failure
                     then j ← j + 1

main
inst : instantiation
failure : Boolean
success ← FALSE
prop(0, D, failure)
if not failure
  then backtrackfreeprop(1, D, success)
```

Algorithm 0.2: BACKTRACKPROPMODULE(C, D)

TYPE: $domains = domain^n$

TYPE: $instantiation = element[n]$

procedure BACKTRACKPROP(j, D, **success**)

while $D[j] \neq \emptyset$ **and** **not** $success$

$$
\textbf{do} \begin{cases} d \leftarrow some(D[j]) \\ D[j] \leftarrow D[j] \setminus \{d\} \\ \textbf{if } const(inst, j, d) \\ \quad \textbf{then} \begin{cases} inst[j] \leftarrow d \\ success \leftarrow (j = n) \\ \textbf{if not } success \\ \quad \textbf{then} \begin{cases} prop(j, D, failure) \\ \textbf{if not } failure \\ \quad \textbf{then } backtrackprop(j+1, D, success) \end{cases} \end{cases} \end{cases}
$$

main

$inst : instantiation$

$failure : Boolean$

$success \leftarrow FALSE$

$prop(0, D, failure)$

if **not** $failure$

  then $backtrackprop(1, D, success)$

Teoreemat

- (8.14) **backtrackfree** kerää $inst$-listan alkuun vastaavan **karsitun** nimiöintipuun vasemmanpuoleisen haaran tiedot ja asettaa $success$-muuttujan arvoksi $TRUE$ jos tämä haara esittää ratkaisua.
- (8.16) **backtrackfreeprop** kerää $inst$-listan alkuun vastaavan **vyörytettävän** nimiöintipuun vasemmanpuoleisen lehden tiedot ja asettaa $success$-muuttujan arvoksi $TRUE$ jos tämä lehti on tasolla $2n$.
- (8.17) Jos ratkaisu on olemassa **backtrack** kerää $inst$-listaan vastaavan **karsitun** nimiöintipuun vasemmanpuoleisen, jotain ratkaisua esittävän haaran, tiedot. Muutoin se ilmoittaa epäonnistumisesta asettamalla $success$-muuttujan arvoksi $FALSE$.
- (8.18) Jos ratkaisu on olemassa **backtrackprop** kerää $inst$-listaan vastaavan **vyörytettävän** nimiöintipuun vasemmanpuoleisen, jotain ratkaisua esittävän lehden, tiedot. Muutoin se ilmoittaa epäonnistumisesta asettamalla $success$-muuttujan arvoksi $FALSE$.