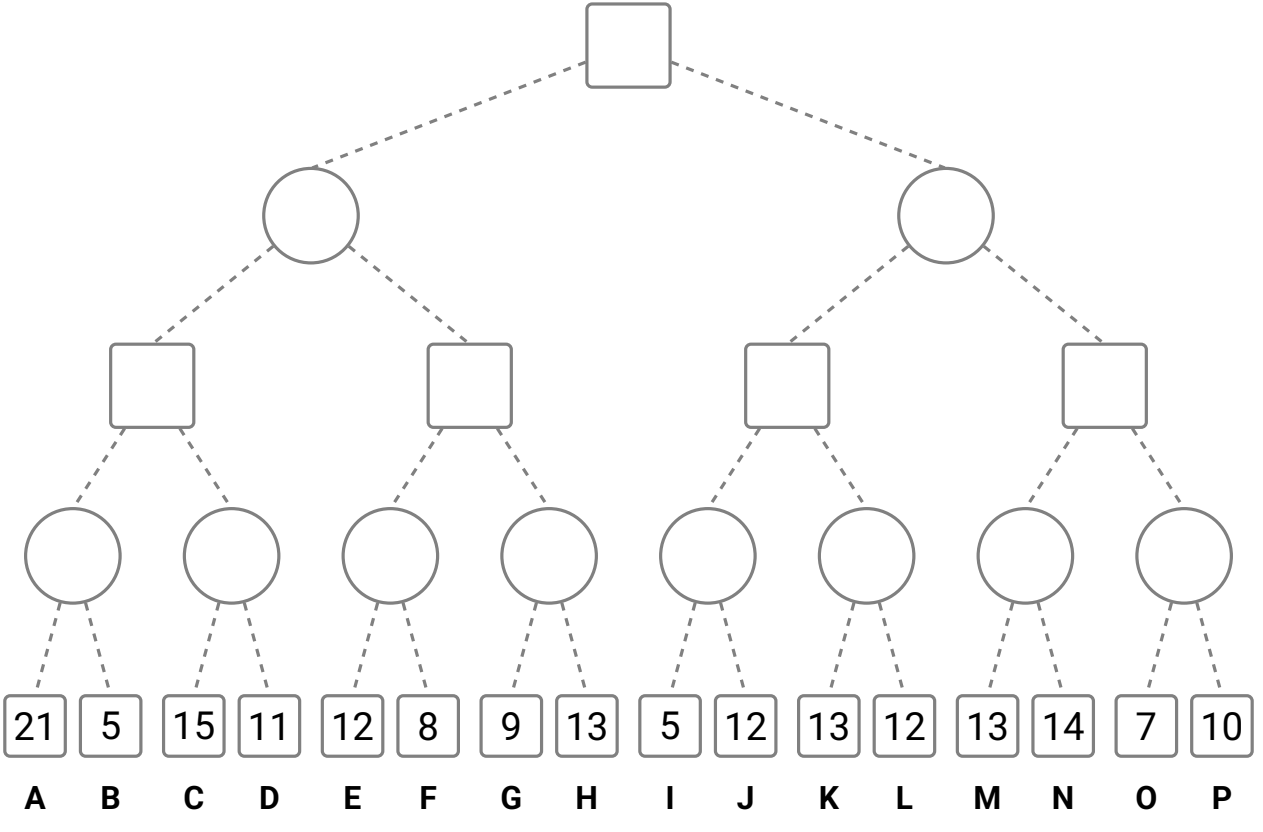
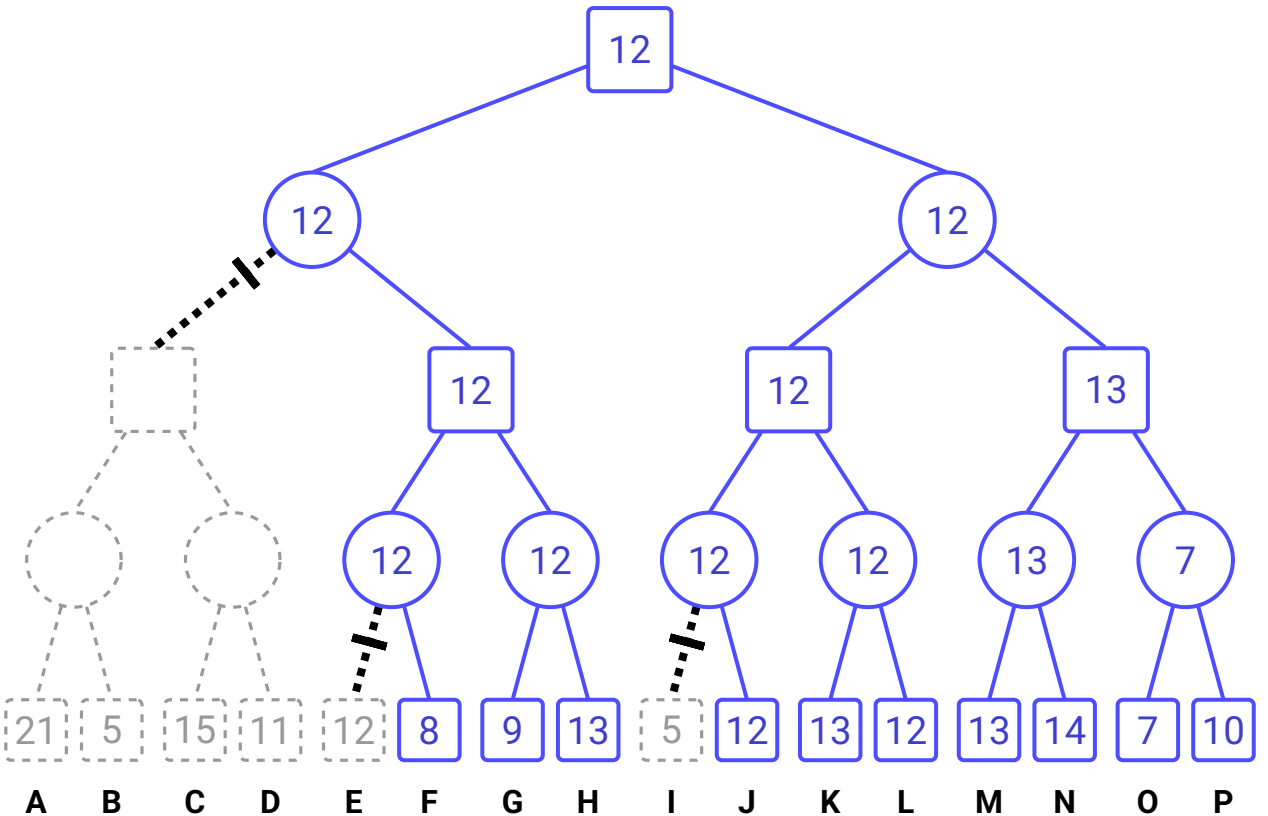
Lecture Game Tree Example — Slide 67

Notes prepared by S. Baskaran

Game Tree – Slide 67



Alpha-Beta Right To Left – Search Tree



Alpha-Beta Right To Left – Solution

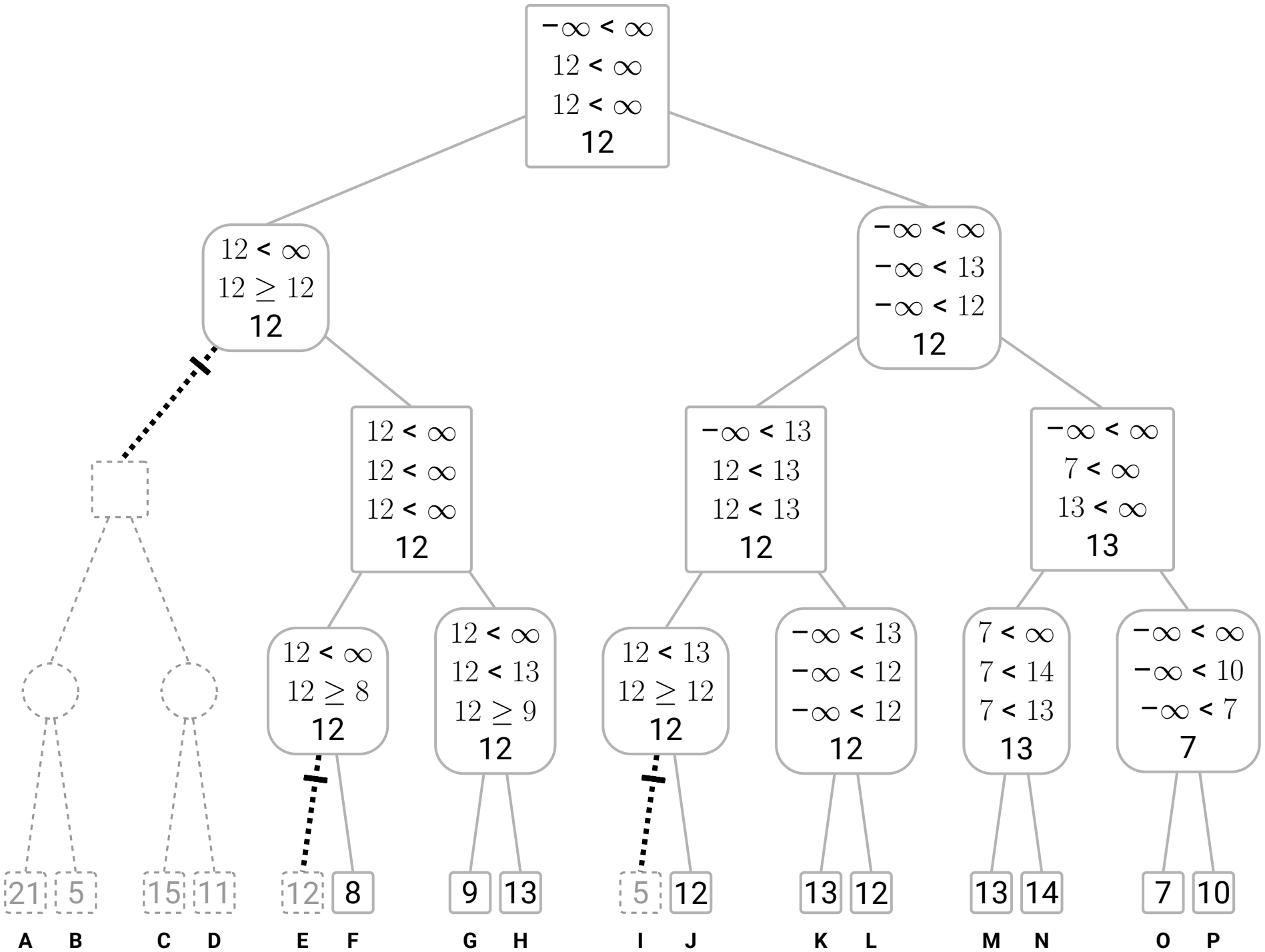
Each non-leaf node displays a list of alpha-beta bounds (open intervals) and a final value.
The bounds are displayed in the format $(\alpha < \beta)$ or $(\alpha \geq \beta)$.

Each non-leaf node displays:

- an initial bound ($\alpha < \beta$) received from the parent node,
- followed by several updated bounds, one for each child inspected,
- and the final value of the node.

The cut-off happens when the interval collapses: $\alpha \geq \beta$.

CALL Alpha-Beta(game-tree, $\alpha = -\infty, \beta = \infty$)



Alpha-Beta Right To Left – Solution

The game tree has 5 levels (a,b,c,d,e,f), the nodes in each level are numbered from left-to-right: the root is a1 followed by b1, b2; then c1,…,c4; d1,…,d8; and e1,…,e16.

Node Info.: (NODE,PLAYER,ALPHA,BETA,EVAL)

```
CALL Alpha-Beta-Right-To-Left(a1,-inf,+inf)

VISIT        (a1,MAX,-inf,inf)
VISIT            (b2,MIN,-inf,inf)
VISIT                (c4,MAX,-inf,inf)
VISIT                    (d8,MIN,-inf,inf)
VISIT                        (e16,MAX,-inf,inf)
SOLVE LEAF                   (e16,MAX,10,inf,10)
UPDATE BETA              (d8,MIN,-inf,10)
VISIT                        (e15,MAX,-inf,10)
SOLVE LEAF                   (e15,MAX,7,10,7)
UPDATE BETA              (d8,MIN,-inf,7)
SOLVE                    (d8,MIN,-inf,7,7)
UPDATE ALPHA         (c4,MAX,7,inf)
VISIT                    (d7,MIN,7,inf)
VISIT                        (e14,MAX,7,inf)
SOLVE LEAF                   (e14,MAX,14,inf,14)
UPDATE BETA              (d7,MIN,7,14)
VISIT                        (e13,MAX,7,14)
SOLVE LEAF                   (e13,MAX,13,14,13)
UPDATE BETA              (d7,MIN,7,13)
SOLVE                    (d7,MIN,7,13,13)
UPDATE ALPHA         (c4,MAX,13,inf)
SOLVE                (c4,MAX,13,inf,13)
UPDATE BETA      (b2,MIN,-inf,13)
VISIT                (c3,MAX,-inf,13)
VISIT                    (d6,MIN,-inf,13)
VISIT                        (e12,MAX,-inf,13)
SOLVE LEAF                   (e12,MAX,12,13,12)
UPDATE BETA              (d6,MIN,-inf,12)
VISIT                        (e11,MAX,13,12,13)
SOLVE LEAF                   (e11,MAX,13,12,13)
UPDATE BETA              (d6,MIN,-inf,12)
SOLVE                    (d6,MIN,-inf,12,12)
UPDATE ALPHA         (c3,MAX,12,13)
VISIT                    (d5,MIN,12,13)
VISIT                        (e10,MAX,12,13)
SOLVE LEAF                   (e10,MAX,12,13,12)
UPDATE BETA              (d5,MIN,12,12)
PRUNE                        (e9,MAX,12,12)
SOLVE                    (d5,MIN,12,12,12)
UPDATE ALPHA         (c3,MAX,12,13)
SOLVE                (c3,MAX,12,13,12)
UPDATE BETA      (b2,MIN,-inf,12)
SOLVE            (b2,MIN,-inf,12,12)
UPDATE ALPHA (a1,MAX,12,inf)
VISIT            (b1,MIN,12,inf)
VISIT                (c2,MAX,12,inf)
VISIT                    (d4,MIN,12,inf)
VISIT                        (e8,MAX,12,inf)
SOLVE LEAF                   (e8,MAX,13,inf,13)
UPDATE BETA              (d4,MIN,12,13)
VISIT                        (e7,MAX,12,13)
SOLVE LEAF                   (e7,MAX,9,13,9)
UPDATE BETA              (d4,MIN,12,9)
SOLVE                    (d4,MIN,12,9,12)
UPDATE ALPHA         (c2,MAX,12,inf)
VISIT                    (d3,MIN,12,inf)
VISIT                        (e6,MAX,12,inf)
SOLVE LEAF                   (e6,MAX,8,inf,8)
UPDATE BETA              (d3,MIN,12,8)
PRUNE                        (e5,MAX,12,8)
SOLVE                    (d3,MIN,12,8,12)
UPDATE ALPHA         (c2,MAX,12,inf)
SOLVE                (c2,MAX,12,inf,12)
UPDATE BETA      (b1,MIN,12,12)
PRUNE                (c1,MAX,12,12)
SOLVE            (b1,MIN,12,12,12)
UPDATE ALPHA (a1,MAX,12,inf)
SOLVE        (a1,MAX,12,inf,12)
```