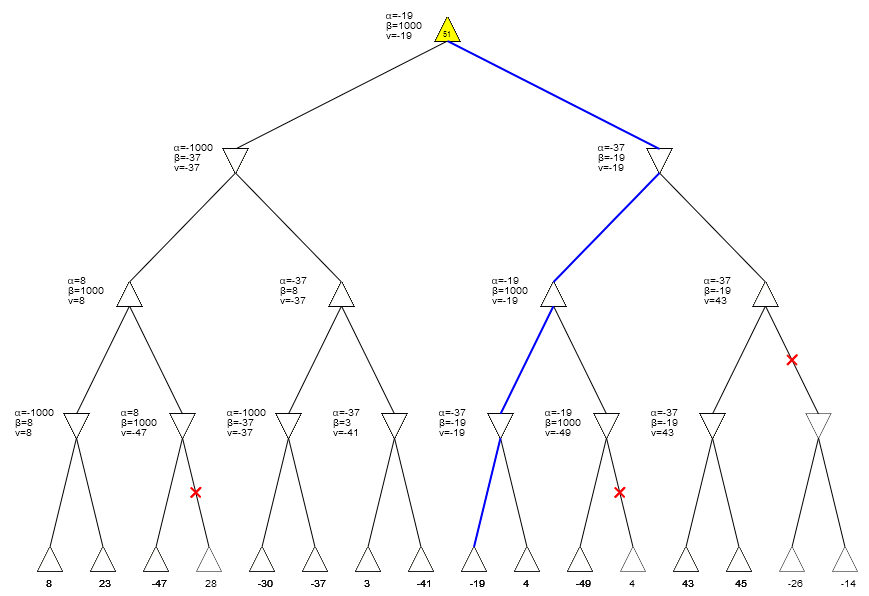
**Heurística**: estimativa

**Utilidade**: análise situação final

Deve ocorrer poda?

* Estamos num nodo MIN de valor X: Se existir algum ancestral MAX maior que X, poda.
* Estamos num nodo MAX de valor X: Se existir algum ancestral MIN menor que X, poda.



Os valores de α e β são passados para os filhos

**Nodo MIN só atualiza β**

**Nodo MAX só atualiza α**

Em nodos MIN com filho de valor X:

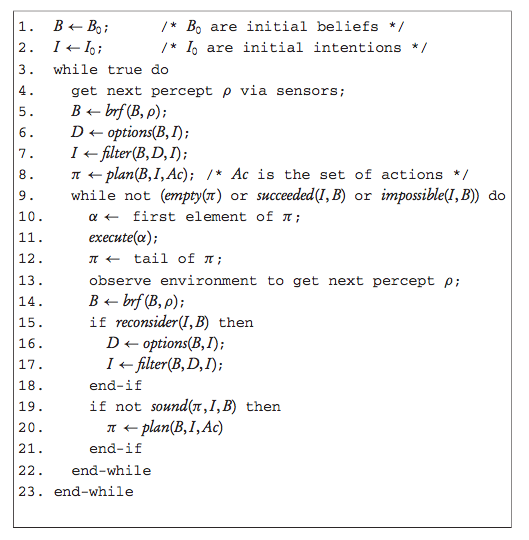
* Se X no intervalo de α e β do nodo MIN, atualiza β do nodo MIN com X
* Ao retornar para o nodo MAX:
  + Se β do MIN foi alterado (foi encontrado X menor que β) e β do MIN for maior que α do MAX, atualiza α do MAX com β do MIN
  + Se β do MIN não foi alterado (não foi encontrado X menor que β), atualiza α do MAX com X

Em nodos MAX com filho de valor X:

* Se X no intervalo de α e β do nodo MAX, atualiza α do nodo MAX com X
* Ao retornar para o nodo MIN:
  + Se α do MAX foi alterado (foi encontrado X maior que α) e α do MAX for menor que β do MIN, atualiza β do MIN com α do MAX
  + Se α do MAX não foi alterado (não foi encontrado X maior que α), atualiza β do MIN com X

**Ontologias**

**Agentes**



Sintaxe do comando Jason:

**evento** : **condição** <- **corpo**;

Eventos

* Refletem as mudanças externas e internas (reatividade)
* Adição de crença (atualização proveniente das percepções do ambiente)

+garbage(r1) : not .desire(carry\_to(r2))

<- !carry\_to(r2).

* Adição de objetivo (irá se tornar um desejo)

+!check(slots) : not garbage(r1)

<- next(slot);

!check(slots).

* Remoção de crença (atualização proveniente das percepções do ambiente)

-garbage(r1)

* Remoção de objetivo (falha, irrelevância, desistência…)

-!carry\_to(r2)

* Instruções que podem ser executadas em paralelo

+!!some\_plan(x)