

对抗搜索

Adversarial Search

Outline

- ❖ 博弈论
- ❖ 极大极小搜索算法
- ❖ α - β 剪枝技术
- ❖ 蒙特卡罗树搜索
- ❖ 随机博弈

一、博弈论

- ❖ 下棋、打牌、战争等一类竞争性智能活动称为**博弈**，一般来说，博弈包括一系列的**玩家**、**动作**、**策略**和最终的**报酬**
- ❖ **玩家**：参与博弈的理性主体。如
 - ◆ 拍卖中的竞标者
 - ◆ 玩石头剪刀布的玩家
 - ◆ 参加选举的政治家等
- ❖ **报酬**：所有玩家在达到某种结果时得到的回报
 - ◆ 可以是积极的，也可以是消极的
 - ◆ 每个主体都是自私的，希望得到最大化的报酬



“深蓝”

1997年5月11日，IBM开发的“**深蓝**”击败了国际象棋冠军卡斯帕罗夫。

卡氏何许人也？

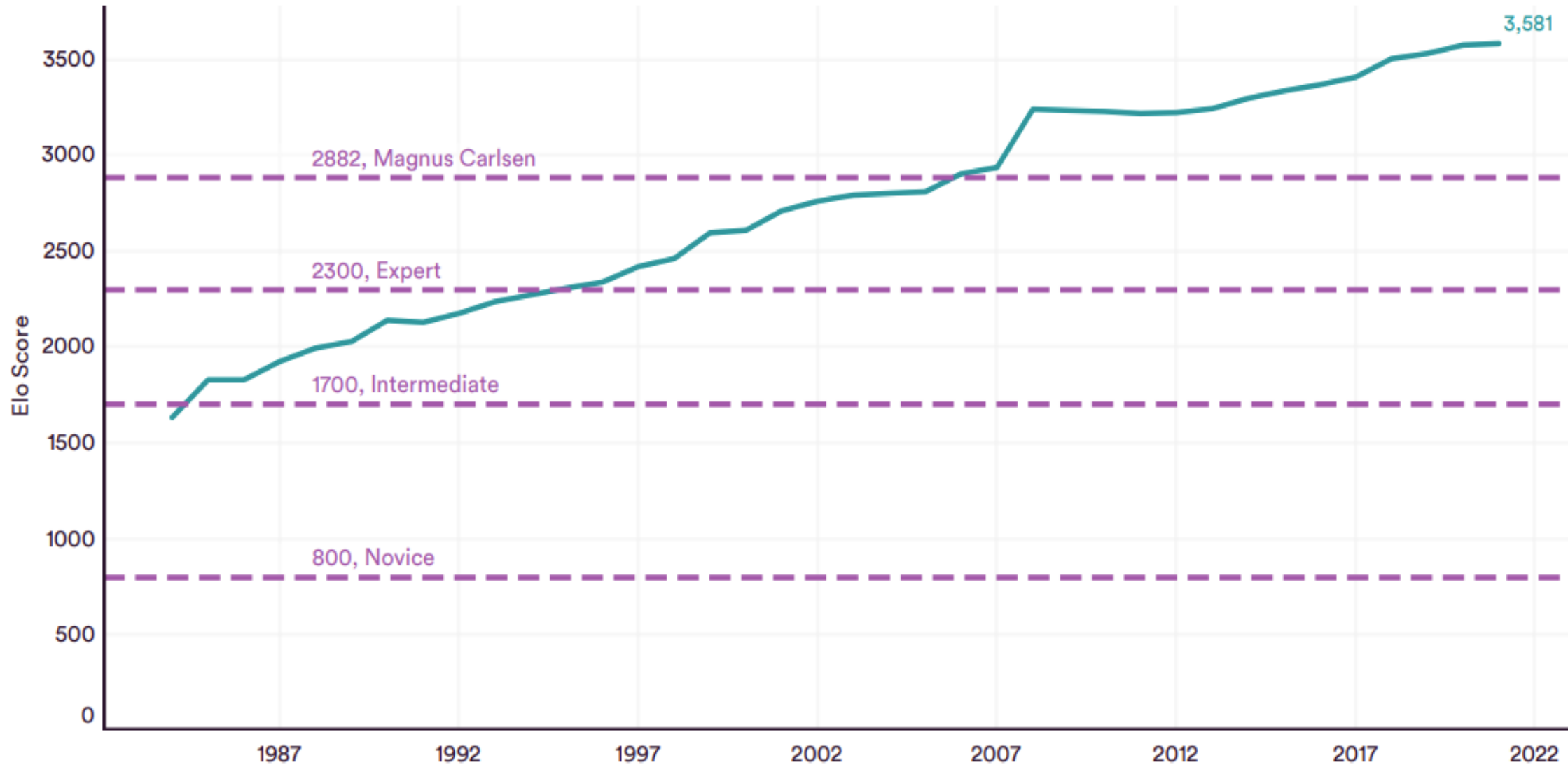
- 1980年他获得世界少年组冠军
- 1982年他并列夺得苏联冠军
- 1985年22岁的卡斯帕罗夫成为**历史上最年轻**的国际象棋冠军。积分是2849，这一分数是有史以来最高分，远远领先于第二位的克拉姆尼克的2770



1997年纽约，与IBM深蓝电脑终局对弈

Chess Software Engines: Elo Score

Source: Swedish Computer Chess Association, 2021 | Chart: 2022 AI Index Report



一个时代的结束

❖ 围棋被认为人类「对抗」计算机的最后壁垒

❖ 2016.3月：AlphaGo 4:1 战胜李世石

感到惊讶——无话可说——令人绝望

❖ 棋盘游戏作为AI进步衡量标尺的时代宣告结束



游戏AI的发展历程

The Development of Gaming AI

非完全信息游戏难度比较

Difficulty of imperfect information games

游戏

两人德州扑克 (限注)
两人德州扑克 (无限注)
桥牌
麻将

信息集数目

10^{14}
 10^{162}
 10^{67}
 10^{121}

信息集平均大小

10^3
 10^3
 10^{15}
 10^{48}

隐藏的
不确定信息
Hidden information



博弈论中的纳什均衡

❖ 囚徒困境




		Ben	
		Silent	Confess
Alan	Silent	A:-1, B:-1	A:-15, B:0
	Confess	A:0, B:-15	A:-10, B:-10

囚徒困境



		Ben confesses	
		Silent	Confess
Alan	Silent	A: -1, B: -1	A: -15, B: 0
	Confess	A: 0, B: -15	A: -10, B: -10

		Ben stays silent	
		Silent	Confess
Alan	Silent	A: -1, B: -1	A: -15, B: 0
	Confess	A: 0, B: -15	A: -10, B: -10



We go to prison
for 10 years because
we both confessed. If
we hadn't confessed,
we would each have
gone to jail for only
one year.

Yeah!
But if I told
you that I wouldn't
confess, you still would
have confessed to avoid
prison. Then I would
have gone to jail for 15
years. I'm glad I
confessed.

双人零和博弈

最常研究的博弈（如国际象棋和围棋），特点：双人零和、全信息、非偶然

❖ 对垒双方(A、B)轮流走步，结果只有三种：A胜B败、A败B胜、双方平局。

二人获得分数的代数和必为零，称为“双人零和”。

❖ 对垒过程中任何一方都了解当前格局及过去的历史。

❖ 任何一方都要根据当前情况，分析得失，选取对自己最有利而对对方最不利的对策，而不存在“碰运气”的偶然因素。
即双方都是很理智地决定自己的行动。



❖ 以某一方的立场把双人完备信息博弈过程用图表示出来，就得到一棵与或树。描述博弈过程的与或树称为**博弈树**。

❖ 博弈树的特点：

- ◆ 博弈的初始格局是初始节点。
- ◆ 在博弈树中，“或”节点和“与”节点逐层交替出现。自己一方扩展的节点之间是“或”关系，对方扩展的节点之间是“与”关系。双方轮流地扩展节点。
- ◆ 所有能使自己获胜的终局都是本原问题，相应的节点是可解节点；所有使对方获胜的终局都是不可解节点。

二、极大极小方法（Minimax algorithm）

- ❖ 设博弈的双方中一方为**A**,另一方为**B**。然后为其中的一方(例如**A**)寻找一个最优行动方案。
 - ◆ 考虑每一方案实施后对方可能采取的所有行动,并计算可能的得分。定义一个估价函数,用来估算当前博弈树端节点的得分。估算出的得分称为**静态估值**。
 - ◆ 由端节点估值推算出父节点的得分——**倒推值**
 - ▣ “或”节点,选其子节点中最大的得分作为父节点的得分,立足最好;
 - ▣ “与”节点,选其子节点中最小的得分作为父节点的得分,立足最坏。
 - ◆ 如果一个行动方案能获得**较大的倒推值**,则它就是当前最好的行动方案。

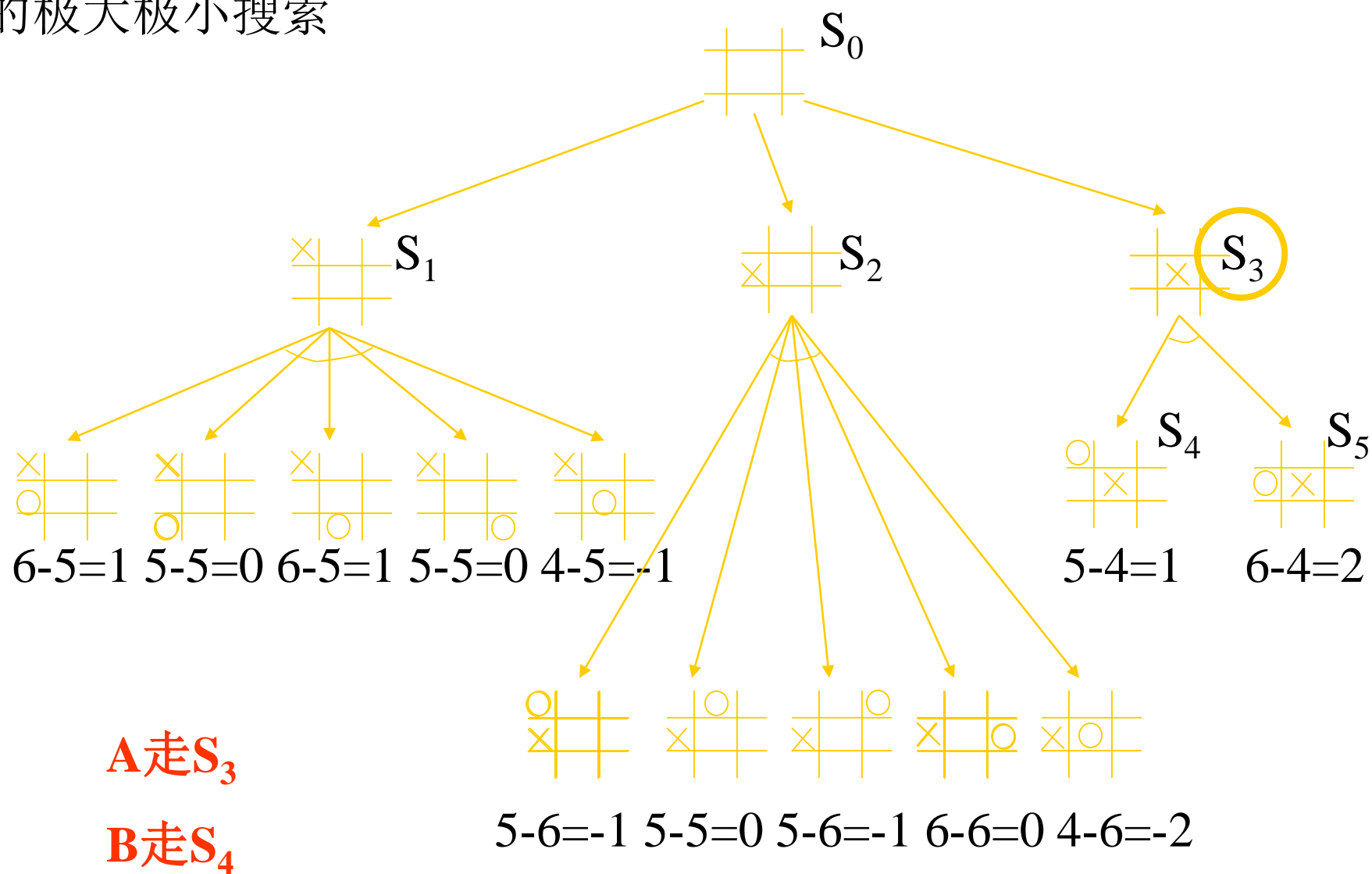
例：一字棋游戏

设有3x3的九个空格, 由A, B二人对弈, 轮到谁走棋谁就往空格上放一只自己的棋子, 先使自己的棋子构成“三子成一线”者胜利。

解：设棋局为 P , 定义估价函数为 $e(P)$:

- ✓ 若 P 是A必胜的棋局, 则 $e(P)=+\infty$;
 - ✓ 若 P 是B必胜的棋局, 则 $e(P)=-\infty$;
 - ✓ 若 P 是胜负未定的棋局, 则 $e(P)=e(+P)-e(-P)$, 其中 $e(+P)$ 表示棋局 P 上有可能使a成为三子成一线的数目; $e(-P)$ 表示棋局 P 上有可能使b成为三子成一线的数目。
- 具有对称性的棋盘认为是同一棋盘。

一字棋的极大极小搜索



❖ 极大极小算法: 对博弈树进行完整的深度优先探索

- ◆ 时间复杂度: $O(b^m)$ (树的最大深度为 m , 每个点都有 b 种移动)
- ◆ 空间复杂度: $O(bm)$ (当一次生成所有动作时)
 $O(m)$ (当一次只生成一个动作时)

✗ 无法应用于复杂博弈

- ◆ 例如: 国际象棋的分支因子约为35, 平均深度约为 80层, 搜索 $35^{80} \approx 10^{123}$ 个状态是不可行的。

✓ 是对博弈进行数学分析的基础。通过以各种方式近似极大极小分析, 可以推导出更实用的算法。

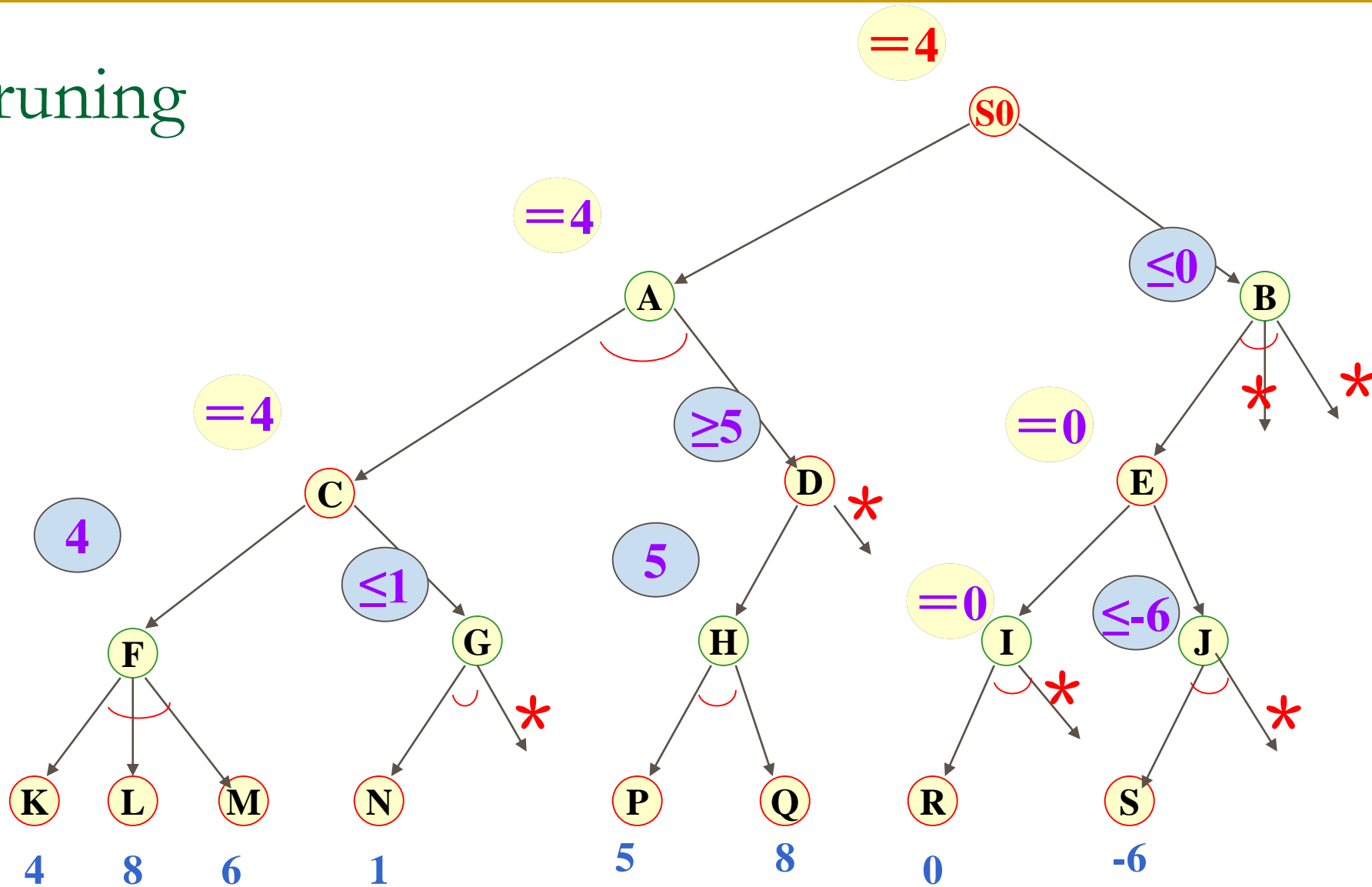
三、 α - β 剪枝 (α - β pruning)

❖ α - β 剪枝技术的**基本思想**:

边生成博弈树边计算评估各节点的倒推值，并且根据评估出的倒推值范围，及时停止扩展那些已无必要再扩展的子节点，即相当于剪去了博弈树上的一些分枝，从而节约了机器开销，提高了搜索效率。

- ❖ 对于 “或” 节点，为了剪除某些分枝，取其子节点中的**最大倒推值**作为当前下界的参考，称此值为 **α 值**；
- ❖ 对于 “与” 节点，应取其子节点中的**最小倒推值**作为当前上界的参考，称此值为 **β 值**。
- ❖ 剪枝技术的一般规律：
 - ◆ 任何或节点n的 α 值，如果不能降低其父节点的 β 值，则对节点n以下的分支可停止搜索，并使n的倒推值为 α 值。
 - ◆ 任何与节点n的 β 值，如果不能升高其父节点的 α 值，则对节点n以下的分支可停止搜索，并使n的倒推值为 β 值。
 - 对于一个或节点，如果估值最高的节点最先生成，或者对于一个与节点，估值最低的子节点最先生成，则被剪的节点数最多，搜索的效率最高，称为**最优 α - β 剪枝**。

α - β pruning



α - β pruning

- ❖ $\alpha - \beta$ 剪枝可以应用于任何深度的树，不仅可以对叶节点进行剪枝，而且也可以对整个子树进行剪枝。
- ❖ $\alpha - \beta$ 剪枝的有效性很大程度依赖于节点的扩展顺序。拥有完美扩展顺序的 $\alpha - \beta$ 剪枝可以求解的树的深度约为极大极小搜索算法的两倍。
 - ◆ 可以用排序函数减少扩展的节点数，如：国际象棋中，先尝试吃子，接着是威胁，然后是前进后退
 - ◆ 增加动态的排序方案，如先尝试已发现的最佳顺序，或者用迭代加深方法进行探索等。

四、复杂棋类游戏常用策略

❖ A型策略

- ◆ 考虑搜索树中某一深度的所有可能的移动，然后使用启发式评价函数对该深度下的节点进行评估。
- ◆ 探索树的宽但浅的部分
- ◆ 大多数国际象棋程序采用A 型策略

❖ B型策略

- ◆ 舍弃看起来就很差的节点，“尽可能”走更有可能的路线。
- ◆ 探索树的深但窄的部分
- ◆ 围棋程序通常采用B 型策略

五、开局和残局的处理——搜索和查表

- ❖ 许多游戏程序使用查表而非搜索来处理开局和残局。
- ❖ 开局：开始时可能的局面很少，大多数局面都能存储在表中
 - ◆ 拷贝人类的开局经验
 - ◆ 从玩过的游戏数据库中收集统计好的开局
- ❖ 移动 **10 ~ 15** 步后，到达一个很少见到的局面时，从查表切换到搜索。
- ❖ 游戏接近结束时，可能的局面变少 → 查表。
 - ◆ 计算机对残局的分析能力远远超过了人类。

六、蒙特卡罗树搜索

(Monte Carlo tree search, MCTS)

- ❖ 对围棋来说，启发式 α - β 树搜索有两个主要缺点：
 - ◆ 围棋的分枝因子开始时为361，这意味着 α - β 搜索被限制在4~5层。
 - ◆ 很难为围棋定义一个好的评价函数。
- 为了应对这两个挑战，现代围棋程序放弃了 α - β 搜索，而是使用一种称为蒙特卡罗树搜索的策略。

基本的 MCTS 策略

- ❖ 不使用启发式评价函数，而是根据从该状态开始的**多次完整博弈模拟**进行估算。
 - ◆ 一次模拟(也被称为一个 **playout** 或 **rollout**)先为一个参与者选择移动，接着为另一个参与者选择，重复上述操作直到到达某个终止局面，并根据博弈规则决定输赢及估值。
 - ◆ 用一个模拟策略(**playout policy**)，使其偏向于好的行动。
 - ▣ 不同游戏使用不同的启发式方法；使用神经网络从自我对弈中学习模拟策略等
 - ◆ 判断从什么局面开始模拟，以及分配给每个局面多少次模拟？
 - ▣ 纯蒙特卡罗搜索：从博弈当前状态开始做**N**次模拟，并记录胜率最高的走步
 - ▣ 选择策略 (**selection policy**): 维护一个搜索树，有选择地将计算资源集中在博弈树的重要部分，每次迭代（包含选择、扩展、模拟、反向传播**4**个步骤）中不断增长搜索树。

博弈搜索算法的局限性

- ❖ 计算复杂博弈中的最优决策时需进行一些假设和近似，都存在局限性，选择哪种算法在一定程度上取决于每种博弈的特征。
 - ◆ α - β 搜索使用启发式评价函数作为近似 \rightarrow 易受到启发式函数近似误差的影响
 - ◆ 蒙特卡罗搜索：通过博弈模拟来进行近似计算，当分支因子较高或评价函数难以定义时首选蒙特卡罗搜索
 - ◆ α - β 搜索和蒙特卡罗搜索都会计算合法节点的值，更好的搜索算法应该选择使用估值高的节点进行扩展
 - ◆ α - β 搜索和蒙特卡罗搜索都是在单步移动的层级上进行所有推理，而人类可以在更抽象的层级上进行推理，会考虑更高层级的目标(例如，诱捕对方的后)，并使用该目标有选择地生成合理的规划。

AlphaGo

❖ 组成部分：

- ◆ 1. 走棋网络（Policy Network），给定当前局面，预测/采样下一步的走棋。
- ◆ 2. 快速走子（Fast rollout），目标和1一样，但在适当牺牲走棋质量的情况下，速度要比1快1000倍。
- ◆ 3. 估值网络（Value Network），给定当前局面，估计是白胜还是黑胜。
- ◆ 蒙特卡罗树搜索（Monte Carlo Tree Search, MCTS），把以上这三个部分连起来，形成一个完整的系统。

AlphaGo/Zero 的核心组件

- ❖ 蒙特卡洛树搜索——内含用于树遍历的 **PUCT** (Upper Confidence Bound applied to trees) 函数的某些变体
- ❖ 残差卷积神经网络——其中的策略和价值网络被用于评估棋局，以进行下一步落子位置的先验概率估算
- ❖ 强化学习——通过自我对弈进行神经网络训练

七、随机博弈(stochastic game)

- ❖ 包含随机因素的博弈

- ❖ 如西洋双陆棋：运气和技巧相结合的随机游戏

- ◆ 如某个局面中，黑方知道可以走什么棋，但不知道白方会掷出什么，因此也不知道白方的合法移动会是什么
- 黑方无法构建如国际象棋和井字棋中的标准博弈树
- 博弈树中除了MAX和MIN节点外，还必须包括机会节点(chance node)
- ◆ 每个机会节点引出的分支表示可能掷出的骰子点数，每个分支都标有掷出的点数及其概率

八、部分可观测博弈

❖ 主要特征：部分可观测性

- ◆ 四国军棋（**Kriegspiel**）：黑白双方只能看到自己的棋子，裁判可看到所有棋子，对比赛进行判断并定期向双方宣布
- ◆ **Battelship**：每个玩家战舰的放置位置对敌人未知
- ◆ **Stratego**：棋子的位置已知，但种类隐藏
- ◆ 纸牌游戏：桥牌、红心大战、扑克等
-
- ◆ 真实战争：敌人行踪是未知的

Q & A
