我们讲一下使用 propositional logic 解决 N-queens problems，我们在这里和课件不同的是，让 N=4，而不是 8，不然接下来的讨论会很辛苦。当 N=2 或者 N=3 的时候，问题是无解的，而 N=1 时问题有显而易见的解（truism）。

要解决这个问题，第一步是对问题进行建模（encoding the problem in propositional logic）。首先，任何 finite computational problem 都可以用二进制数字（binary numbers）进行 encode，也就是说可以用 truth values 进行 encode。假设，此时我想让一个 variable p 取值为 1,2,3 中的一个。那么可以用 p1 p2 p3 作为 atoms 来完成这个任务，如果 p1 为真，则表示 p 取值为 1，p2 p3 同理。那么，命题逻辑表达式 p1 or p2 or p3 表示 p 至少取值为 1,2,3 中的一个，否则整个式子就会取值为 0，这个好理解吧？在看下面一个表达式：(~p1 or ~p2) and (~p1 or ~p3) and (~p2 or ~p3)表示 p 最多可以取值为 1,2,3 中的一个，不可以取值为多于一个，比如同时取值为 p=1 和 p=2 是不可以的。为什么呢？因为如果让 p1 为真，则~p1 为假，所以~p2 和~p3 就必须为真，反过来 p2 和 p3 就必须为假，只有这样，整个表达式才可以为真。那么 p1 or p2 or p3 和（~p1 or ~p2) and (~p1 or ~p3) and (~p2 or ~p3)这两个式子交在一起可以表示：p 取值为 1,2,3 中的一个且仅一个（英文：p must have exactly one of the values）。有了这个思路，我们看 4-queens problems 怎么 encode。

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44

在一个 4*4 的棋盘里，我们需要 16 个 atoms 来对问题进行 encode：qij，其中 1≤i≤n, 1≤j≤n (n=4)。我们用 qij 来表示一个 queen 被放置在了 row i 和 column j 的位置。下面的表达式可以表示：at least one queen is placed in each row（一共 4 个 clauses）。

q11 or q12 or q13 or q14，
q21 or q22 or q23 or q24，
q31 or q32 or q33 or q34，
q41 or q42 or q43 or q44。

上述表达式的每一行，就是课件中的 $QueenPlaced_i = q_{i1}$ or···or q_{in}。四行合在一起就是课件中的 $NQueensPlaced = And_i QueenPlaced_i$。

根据 n-queens problem 的规则，要想让 queens 之间不互相攻击，需要满足三个条件：每一行，每一列，每一个对角线都不能放置两个 queens。

1. no more than one queen is placed in each row（一共 24 个 clauses）：

（1）~q11 or ~q12, ~q11 or ~q13, ~q11 or ~q14, ~q12 or ~q13, ~q12 or ~q14, ~q13 or ~q14
也就是~q_{1j} or ~$q_{1,j+k}$，其中 1≤j≤n，1≤k≤n-j；或者写为 $q_{1j}\rightarrow$~$q_{1,j+k}$ 。
表示：如果 q_{1j} 位置放置了一个 queen，则 $q_{1,j+k}$ 不可以再放置一个 queen。

同样的道理：

（2）~q21 or ~q22, ~q21 or ~q23, ~q21 or ~q24, ~q22 or ~q23, ~q22 or ~q24, ~q23 or ~q24
（3）~q31 or ~q32, ~q31 or ~q33, ~q31 or ~q34, ~q32 or ~q33, ~q32 or ~q34, ~q33 or ~q34
（4）~q41 or ~q42, ~q41 or ~q43, ~q41 or ~q44, ~q42 or ~q43, ~q42 or ~q44, ~q43 or ~q44

可通用表示为：$q_{ij} \rightarrow \sim q_{i,j+k}$
以上合在一起，对应我们课件中的 Rij 条件。

2. no more than one queen is placed in each column（一共 24 个 clauses）：

（1）~q11 or ~q21, ~q11 or ~q31, ~q11 or ~q41, ~q21 or ~q31, ~q21 or ~q41, ~q31 or ~q41
也就是$\sim q_{i,1}$ or $\sim q_{i+k,1}$，其中 $1 \leqslant i \leqslant n$，$1 \leqslant k \leqslant n-i$；或者写为 $q_{i,1} \rightarrow \sim q_{i+k,1}$ 。
表示：如果 q_{i1} 位置放置了一个 queen，则 $q_{i+k,1}$ 不可以再放置一个 queen。

同样的道理：
（2）~q12 or ~q22, ~q12 or ~q32, ~q12 or ~q42, ~q22 or ~q32, ~q22 or ~q42, ~q32 or ~q42
（3）~q13 or ~q23, ~q13 or ~q33, ~q13 or ~q43, ~q23 or ~q33, ~q23 or ~q43, ~q33 or ~q43
（4）~q14 or ~q24, ~q14 or ~q34, ~q14 or ~q44, ~q24 or ~q34, ~q24 or ~q44, ~q34 or ~q44

可通用表示为：$q_{ij} \rightarrow \sim q_{i+k,j}$
以上合在一起，对应我们课件中的 Cij 条件。

3. no more than one queen is placed in each diagonal（一共 28 个 clauses）：

这个比起前两个条件有点难，我们还是找一个简单的思路。我们的任务是对于棋盘上任何一个格子（cell 或者 square），只要那个位置放置了 queen，则它的对角线位置就不该放置另一个 queen 了，对吧？那我们就可以从 square (1,1)这个位置开始，一旦 q11 成立，则 q22,q33,q44 就不可以放置了，这是(1,1)的唯一一条对角线。但是这不是通用的情况，因为对于 square(2,1)这个位置，有两条对角线，分别是 q23,q34 和 q12。所以通用的情况应该是两条对角线。

那么我们让 square 为(i,j)，如果我们的初始坐标是(1,1)，也就是左上方(top left)那个 square，则对角线位置应该是（i+1,j+1）和（i+1,j-1），继续（i+2,j+2）和（i+2,j-2）直到“碰壁”（角标大于 4 或者小于 1）。这里，可能有同学问，我们不需要找出一个 square 上面的对角线的 squares 么？这个很好理解吧，我们是从左上方（1,1）出发然后进行扫荡式 enumerate，则只需要考虑下方 squares 即可，对吧？如果我们假设从下方 square 初始，则只需要考虑上方的对角线位置的 squares。无论哪个点初始都可以。因为，根据交换律，~q12 or ~q21 ≡ ~q21 or ~q12，也就是说 q12→~q21 ≡ q21→~q12。我们的讲义是 diagonal up right 和 diagonal up left，也就是从底下的 square 进行推进的，这是因为讲义棋盘格子的标号是从左下角（down left）开始的。这里我们按照正常从左往右，从上到下的顺序进行推进：

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44

（1）q11 为真（也就是（1,1）位置放置 queen），则~q11 or ~q22, ~q11 or ~q33, ~q11 or ~q44
（2）q12 为真，则~q12 or ~q21, ~q12 or ~q23, ~q11 or ~q34
（3）q13 为真，则~q13 or ~q22, ~q13 or ~q31, ~q13 or ~q24
（4）q14 为真，则~q14 or ~q23, ~q14 or ~q32, ~q14 or ~q41
…
(12) q34 为真，则~q34 or ~q43

第四行的所有 squares 都不需要讨论，因为既然是向下推进，第四行之下已经没有 row 了，所以一共是上述 12 种情况。

我们用表达式分别对上面的第一、二、三行的情况进行表示：
第一行四个 squares（上面的（1）（2）（3）（4））diag. down right 方向可以写为：
$q_{1j} \rightarrow \sim q_{1+k,j+k}$，其中 $1 \leqslant j \leqslant n$，$1 \leqslant k \leqslant \min\{n-j,n-1\}$
同理，
第二行：$q_{2j} \rightarrow \sim q_{2+k,j+k}$，其中 $1 \leqslant j \leqslant n$，$1 \leqslant k \leqslant \min\{n-j,n-2\}$
第三行：$q_{3j} \rightarrow \sim q_{3+k,j+k}$，其中 $1 \leqslant j \leqslant n$，$1 \leqslant k \leqslant \min\{n-j,n-3\}$
合在一起，写为：$q_{ij} \rightarrow \sim q_{i+k,j+k}$ 其中 $1 \leqslant i \leqslant n$，$1 \leqslant j \leqslant n$，$1 \leqslant k \leqslant \min\{n-i,n-j\}$，也就是对应课件中的 DRU_{ij}。

第一行四个 squares 的 diag. down left 方向可以写为：
$q_{1j} \rightarrow \sim q_{1+k,j-k}$，其中 $1 \leqslant j \leqslant n$，$1 \leqslant k \leqslant \min\{n-1,j-1\}$
同理，
第二行：$q_{2j} \rightarrow \sim q_{2+k,j-k}$，其中 $1 \leqslant j \leqslant n$，$1 \leqslant k \leqslant \min\{n-2,j-1\}$
第三行：$q_{3j} \rightarrow \sim q_{3+k,j-k}$，其中 $1 \leqslant j \leqslant n$，$1 \leqslant k \leqslant \min\{n-3,j-1\}$
合在一起，写为：$q_{ij} \rightarrow \sim q_{i+k,j-k}$ 其中 $1 \leqslant i \leqslant n$，$1 \leqslant j \leqslant n$，$1 \leqslant k \leqslant \min\{n-i,j-1\}$，也就是对应课件中的 DLU_{ij}。

以上所有的 clauses 的交集就是我们最终的 encoding，一共是 4+24+24+28=80 clauses。可以得到 Lemma: 4-queens problems has a solution iff the set of the above 80 clauses is satisfiable.

接下来，我们要对这 80 个 clauses 进行 SAT 验证😊我们采用最高效的 DPLL 算法。首先观察到没有 unit，so we have to decide！让 q11 为真，则~q11 为假，则所有包含 q11 的 clauses 按照规则都会被删掉，同时包含~q11 的 clauses 中的~q11 也会被删掉，而后者可能会产生新的 unit。这个例子中，这一步产生了~q12, ~q13, ~q14, ~q21, ~q31, ~q41, ~q22, ~q33, ~q44 这些。可以看到，仅这一步，就产生了多达 9 个 units。用这种方式继续做下去，80 个 clauses 减少到以下 14 个：

q23 or q24, q32 or q34, q42 or q43
~q23 or ~q24, ~q32 or ~q34, ~q42 or ~q43
~q32 or ~q42, ~q23 or ~q43, ~q24 or ~q34
~q24 or ~q42, ~q23 or ~q32, ~q23 or ~q34, ~q32 or ~q43, ~q34 or ~q43

继续，decide p23 为真，则~p23 为假，这一步产生了 4 个 units：~q24, ~q43, ~q32, ~q34，再

以这些 units 做下去，最后得到 q34, q42 两个 units。而 q34 和上面的~q34 是不可能同时为真的，所以这 80 个 clauses 仅仅通过两步的 decide（decide q11 和 q23 为真）就得到整个 80 个 clauses 为假的结果，也就是说，仅仅这两步，就帮我们从 2^{16} 种可能性中排除了其中的 2^{14} 种。

接下来，按照规则，应该是对于 q11 和 q23 的取值进行 backtrack。这里我们观察会发现，q11 为真是不可能能够满足这 80 个 clauses 的，因为 q11 在左上角，让它为真会产生一种后果，那就是 9 个 squares 不可以放 queen（row 3 个，column 3 个，diagonal 3 个），那么棋盘上就只剩下 6 个空的 squares（见下图），要放 3 个 queens，无论怎么放都至少有两个 queens“接触”。

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44

所以这种作弊方法可以让我们直接 backtrack 到 q11 的取值，而略过 q23 的取值。我们让 q12 为真，则也会产生新的 units，再去做接下来的步骤，最后得到结论，当 q12, q24, q31, q43 为真的时候，80 个 clauses 都为真：

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44

这就是 q12 为真的唯一结果。当然 decide q13 为真可能有别的结果（q14 不可以分配为真，因为在角上）。

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44

而上述两个结果是 4-queens problems 的唯二两组解，由于对称/镜像关系，其实本质上就只有一组解。这就是使用 DPLL 解决 n-queens problems 的思路和全过程。想一想，有没有更加优化的算法？或者在 DPLL 基础上还有没有哪些升级可以做让该算法更高效？