A Practical Approach to Solving Hard Graph Problems

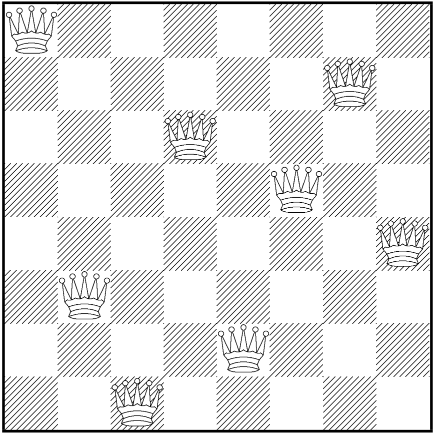
Peter Rossmanith

Theory Group, RWTH Aachen University

September, 24th, 2013

The Eight Queens Problem

How would you solve this?



```
#define N 8
int x[N], c[N], d1[2*N-1], d2[2*N-1];

int main(void) {
   int k = 0, i;
   while(k >= 0) {
      i = x[k];
      if(i == N) k--, i = x[k]++, c[i] = d1[k-i+N-1] = d2[i+k] = 0;
      else if(!c[i] &&!d1[k-i+N-1] &&!d2[i+k])
         if(k == N-1) {
            for(i = 0; i <= k; i++) printf("%d ", x[i]);
            printf("\n");
            x[k]++;
         }
         else c[i] = d1[k-i+N-1] = d2[i+k] = 1, k++, x[k] = 0;
      else x[k]++;
   }
   return 0;
}
```

Alternative solution

Use a CNF-formula to model the 8-queens problem.
Solve it with a SAT-solver.

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} x_{ij} \wedge \bigwedge_{i,j=1}^{n} \Bigg(\bigwedge_{k=i+1}^{n} (\neg x_{ij} \vee \neg x_{kj}) \wedge$$

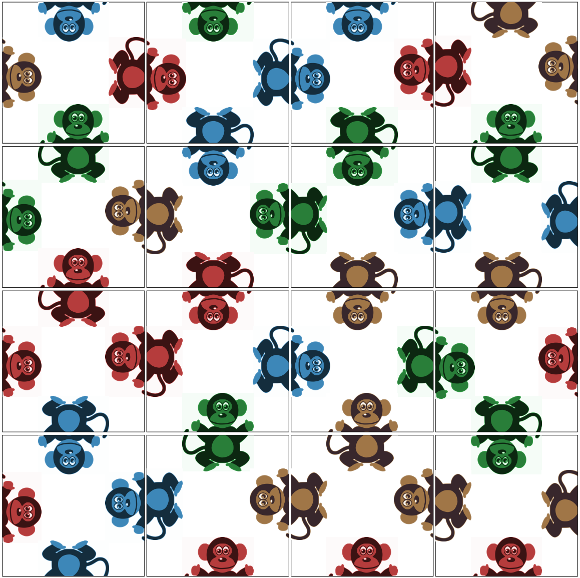
$$\bigwedge_{k=j+1}^{n} (\neg x_{ij} \vee \neg x_{ik}) \wedge$$

$$\bigwedge_{k=1}^{\min\{n-i, n-j\}} (\neg x_{ij} \vee \neg x_{i+k,j+k}) \wedge$$

$$\bigwedge_{k=1}^{\min\{n-i, j-1\}} (\neg x_{ij} \vee \neg x_{i+k,j-k}) \Bigg)$$

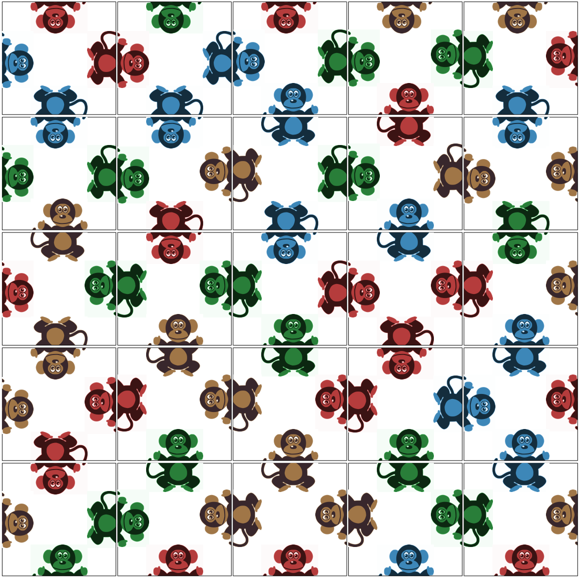
The Monkey Puzzle

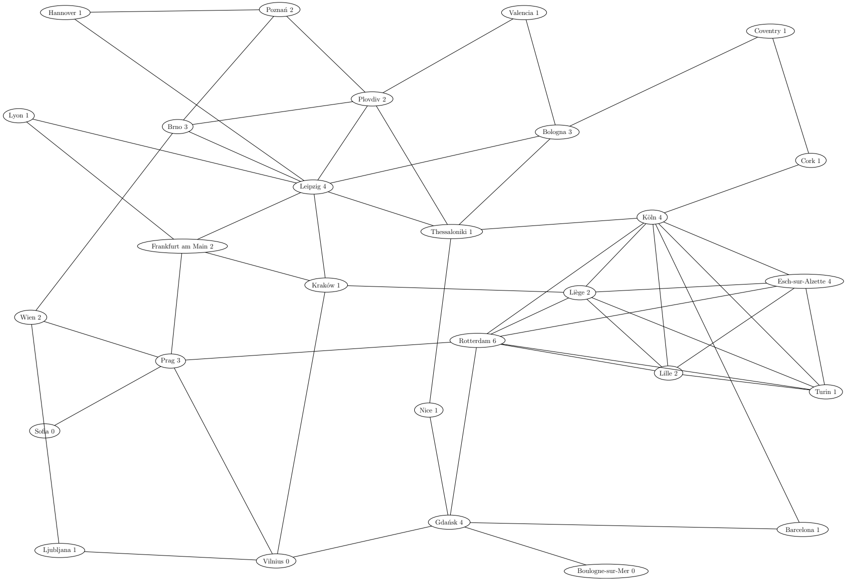
Another combinatorical problem



The Monkey Puzzle

Another combinatorical problem

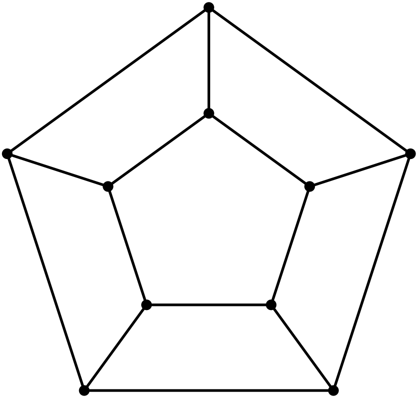




Hannover 1
Poznań 2
Valencia 1
Coventry 1
Plovdiv 2
Lyon 1
Brno 3
Bologna 3
Cork 1
Leipzig 4
Köln 4
Thessaloniki 1
Frankfurt am Main 2
Kraków 1
Esch-sur-Alzette 4
Liège 2
Wien 2
Rotterdam 6
Prag 3
Lille 2
Turin 1
Nice 1
Sofia 0
Gdańsk 4
Barcelona 1
Ljubljana 1
Vilnius 0
Boulogne-sur-Mer 0

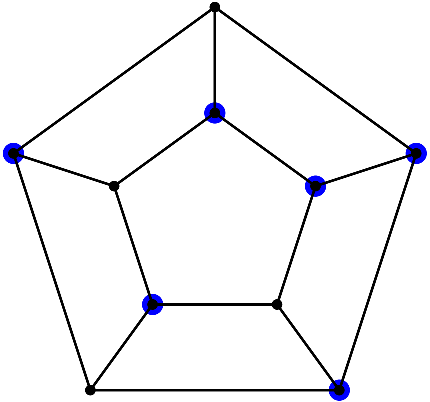
RWTHAACHEN

Example: Vertex Cover



Graph Problems

Example: Vertex Cover



Expressing Vertex Cover as an ILP

Let $G = (V, E)$ be a graph with $V = \{v_1, \ldots, v_n\}$.

$$\begin{aligned} \text{Minimize } & v_1 + \ldots + v_n \\ \text{subject to } & 0 \leq v_i \leq 1 \text{ for } i = 1, \ldots, n \\ & v_i + v_j \geq 1 \text{ for } \{v_i, v_j\} \in E \\ & v_i \in \mathbf{Z} \text{ for } i = 1, \ldots, n \end{aligned}$$

Every NP-complete problem can be reduced to an ILP (but often it is a bad idea to do so).

Expressing Vertex Cover in MSO

MSO = monadic second order logic

C is a vertex cover iff

$$\forall x \forall y (adj(x, y) \rightarrow (x \in C \vee y \in C)).$$

Goal: Find a C of minimal size.

Another example: 3-colorability

$$\begin{aligned}
\exists R \exists G \exists B \, (& \forall x (x \in R \vee x \in G \vee x \in B) && (1)\\
& \wedge \forall x \forall y \, adj(x, y) \rightarrow (x \notin R \vee x \notin G) && (2)\\
& \wedge \forall x \forall y \, adj(x, y) \rightarrow (x \notin R \vee x \notin B) && (3)\\
& \wedge \forall x \forall y \, adj(x, y) \rightarrow (x \notin G \vee x \notin B) \,) && (4)
\end{aligned}$$

Expressing Vertex Cover in MSO

MSO = monadic second order logic

C is a vertex cover iff

$$\forall x \forall y (adj(x, y) \rightarrow (x \in C \vee y \in C)).$$

Goal: Find a C of minimal size.

Another example: 3-colorability

$$\begin{aligned}
\exists R \exists G \exists B \, (\, \forall x (x \in R \vee x \in G \vee x \in B) & \qquad (1) \\
\wedge \, \forall x \forall y \, adj(x, y) \rightarrow (x \notin R \vee x \notin G) & \qquad (2) \\
\wedge \, \forall x \forall y \, adj(x, y) \rightarrow (x \notin R \vee x \notin B) & \qquad (3) \\
\wedge \, \forall x \forall y \, adj(x, y) \rightarrow (x \notin G \vee x \notin B) \,) & \qquad (4)
\end{aligned}$$

Courcelle's Theorem

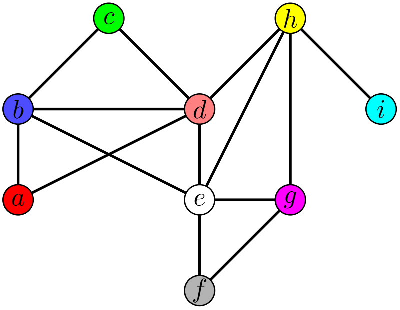
Theorem (Courcelle, 1990)

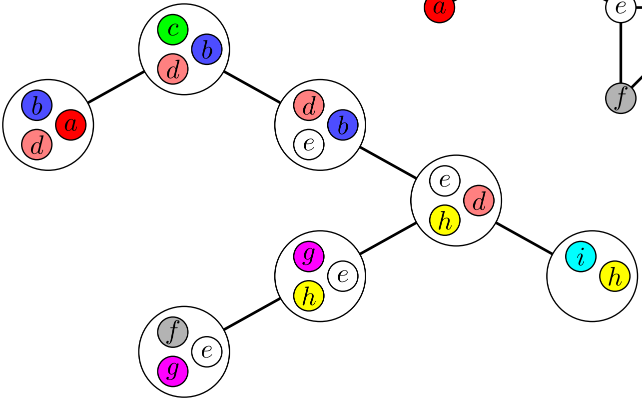
Every problem definable in (Counting) Monadic Second-Order logic (MSO) can be solved in linear time on graphs of bounded treewidth.

Extensions:

- MSO + evaluation relations [Arnborg, Lagergren, Seese, 1991]
- MSO + semiring homomorphisms [Courcelle, Mosbah, 1993]

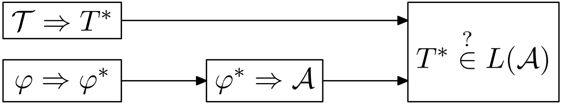
Tree Decompositions and Tree Width





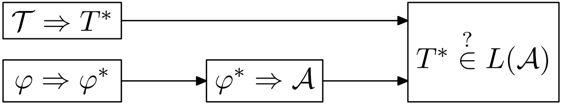
The automata-theoretic approach

Given MSO formula φ and tree decomposition $\mathcal{T}$:



1. construct labeled binary tree T^*, modify φ accordingly
2. construct tree automaton $\mathcal{A}$ with $L(\varphi^*) = L(\mathcal{A})$
3. test whether $\mathcal{A}$ accepts T^*

The automata-theoretic approach



Pro:

- Well-understood, in particular the minimization algorithm.
- Existing optimized tools available (MONA).

Con:

- (Intermediate) automata might be too large to construct.

The automata-theoretic approach

[Niedermeier, "Invitation to Fixed Parameter Algorithms" (2006)]

> *"It must be emphasized, however, that the now described methodology is of purely theoretical interest because the associated running times suffer from huge constant factors and combinatorial explosions [. . .]"*

[Gottlob, Pichler, Wei, 2008]

- MONA: *" 'out-of-memory' already for really small input data."*
- FTA: *"attempt failed with a 'state explosion' "*

Carrère, Kanté, Soguet [Soguet, 2008]:

- $\mathit{DisjointPath}(s_1, s_2, t_1, t_2)$ and clique-width ≤ 2
- *3Col* and $cw \leq 3$
- *Connectedness*, $\mathit{ConnComp}(X)$, $\Delta(G) \leq 2$, . . . and $cw \leq 3$

The automata-theoretic approach

[Gottlob, Pichler, Wei, 2008]

$$\begin{tikzcd}
\boxed{\mathcal{T} \Rightarrow T^*} \arrow[rr] & & \boxed{T^* \overset{?}{\in} L(\mathcal{A})} \\
\boxed{\varphi \Rightarrow \varphi^*} \arrow[r] & \boxed{\varphi^* \Rightarrow \mathcal{A}} \arrow[ur] &
\end{tikzcd}$$

Carrère, Kanté, Soguet [Soguet, 2008]:

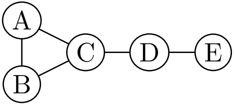
- $\mathit{DisjointPath}(s_1, s_2, t_1, t_2)$ and $cw \leq 2$
- *3Col* and $cw \leq 3$
- *Connectedness*, $\mathit{ConnComp}(X)$, $\Delta(G) \leq 2$, … and $cw \leq 3$

A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\vee\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$

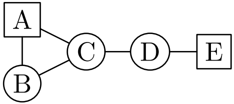
A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$



A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\; \vee \;\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$

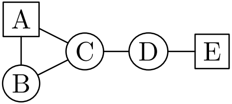


green=true　　　　red=false

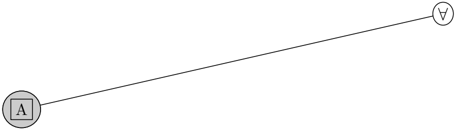
$\forall$

A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

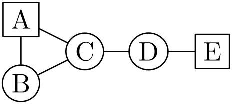


green=true red=false

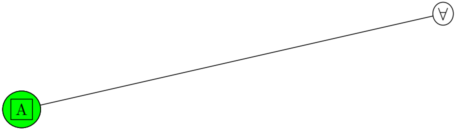


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x, y))\ )$$

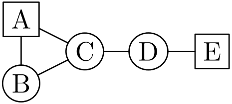


green=true   red=false

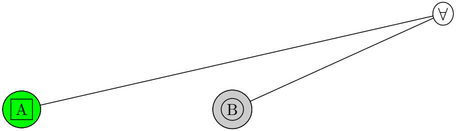


A Game with Two Players

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

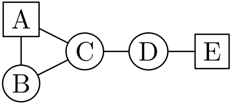


green=true red=false

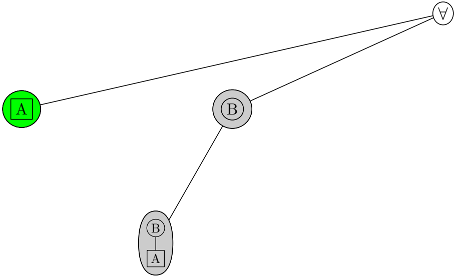


A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\; \vee \;\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$

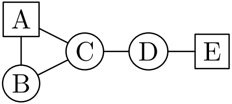


green=true red=false

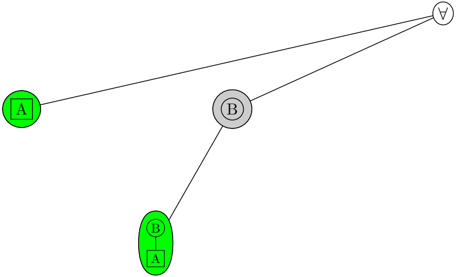


A Game with Two Players

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

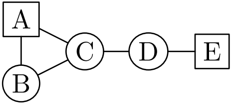


green=true red=false

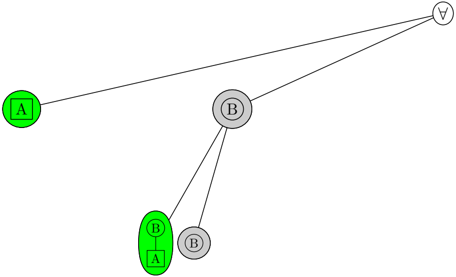


A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\; \vee \;\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$

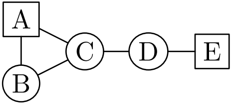


green=true red=false

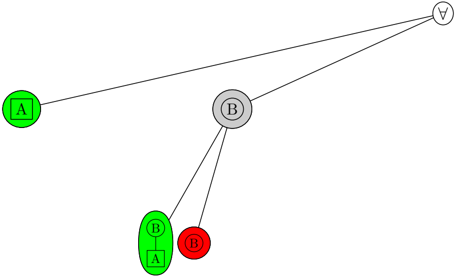


A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\vee\; \exists y \, (y \in U \wedge adj(x,y)) \;)$$

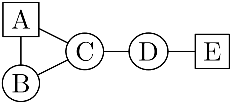


green=true red=false

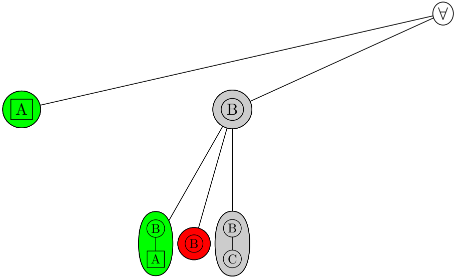


A Game with Two Players

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

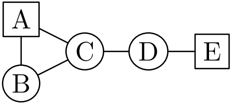


green=true red=false

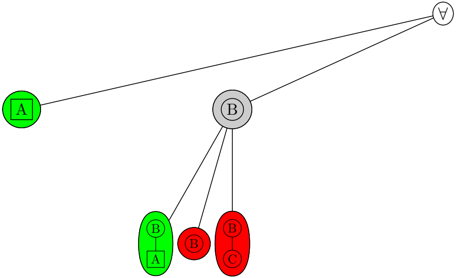


A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\vee\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$

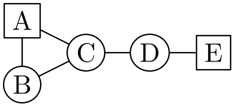


green=true red=false

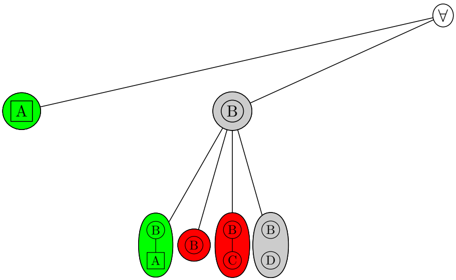


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

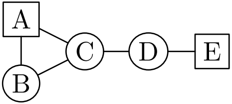


green=true red=false

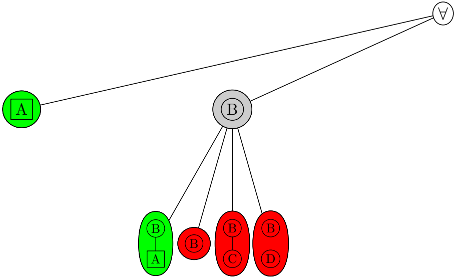


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \ \vee\ \ \exists y\,(y \in U \wedge adj(x,y))\ )$$

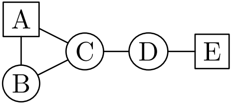


green=true red=false

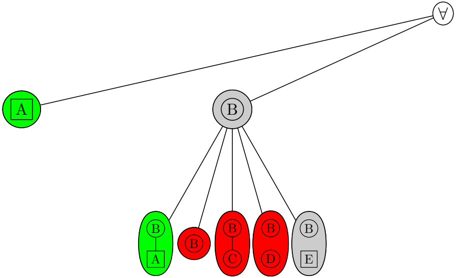


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

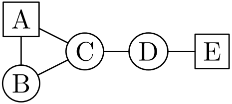


green=true red=false

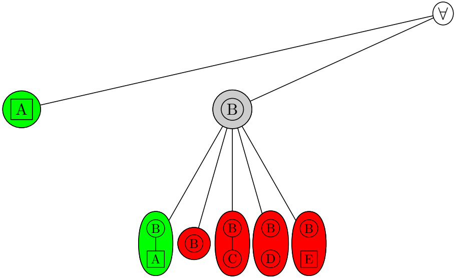


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$

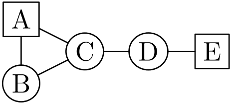


green=true red=false

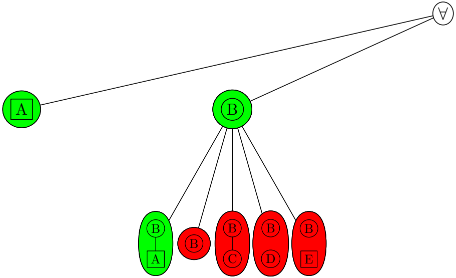


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$

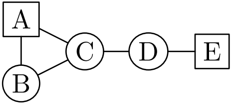


green=true red=false

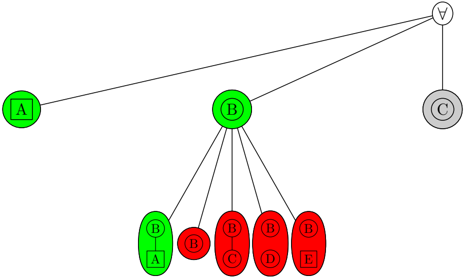


A Game with Two Players

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

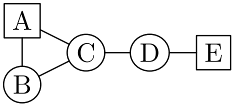


green=true red=false

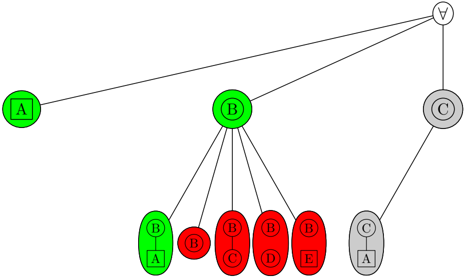


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y \,(y \in U \wedge adj(x,y))\;)$$

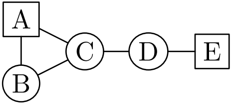


green=true red=false

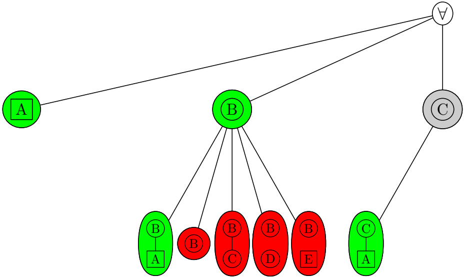


A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\vee\; \exists y \, (y \in U \wedge adj(x,y)) \;)$$

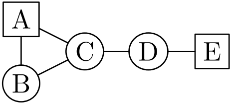


green=true red=false

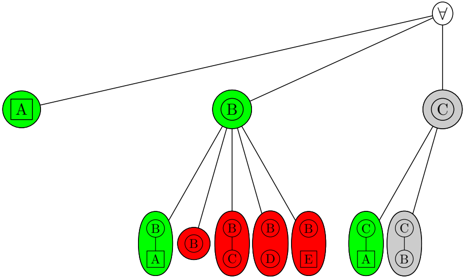


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

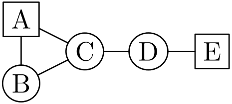


green=true red=false

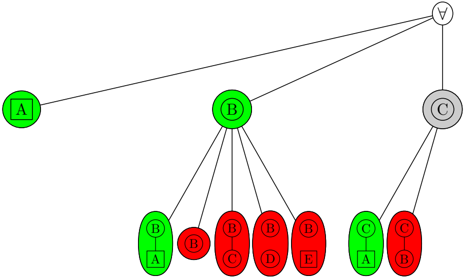


A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\; \vee \;\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$

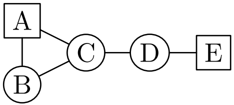


green=true red=false

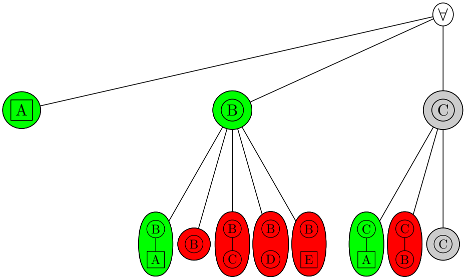


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

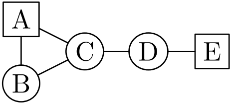


green=true red=false

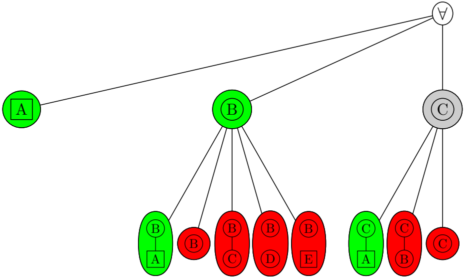


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$

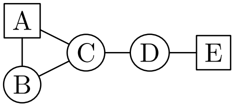


green=true red=false

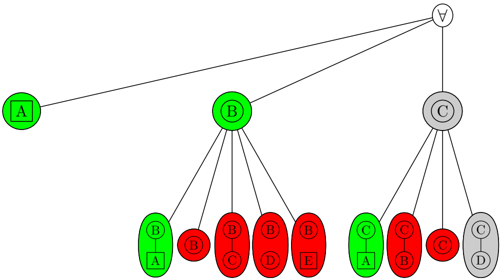


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y \,(y \in U \wedge adj(x,y))\;)$$

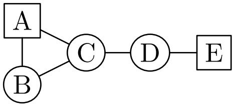


green=true red=false

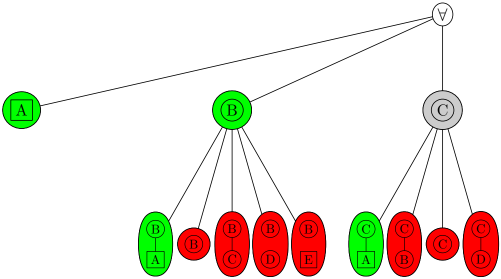


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$

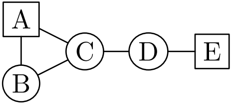


green=true red=false

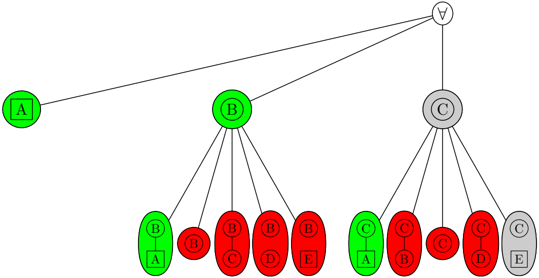


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

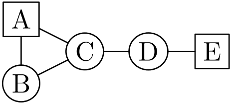


green=true red=false

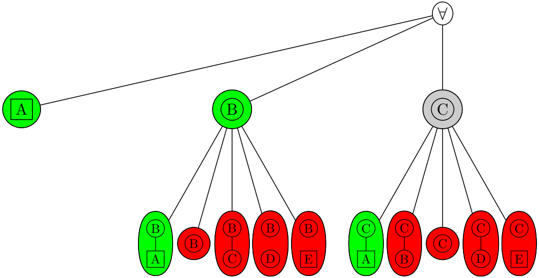


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$

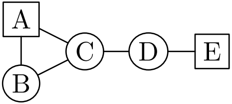


green=true red=false

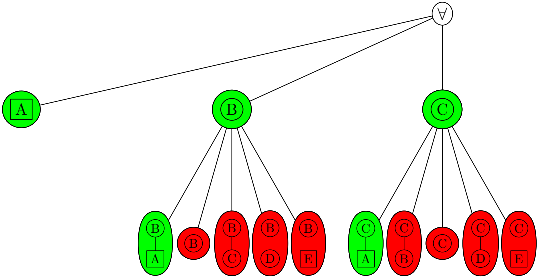


A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \;\vee\; \exists y \, (y \in U \wedge adj(x,y)) \;)$$

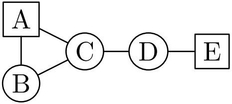


green=true red=false

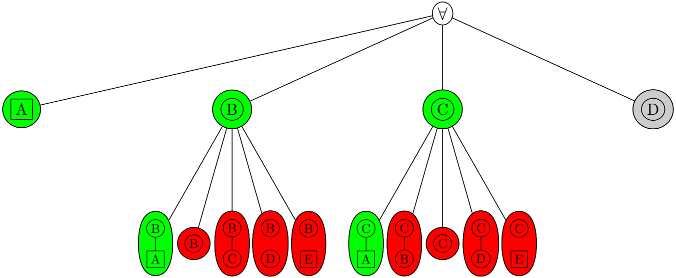


A Game with Two Players

$$ds(U) = \forall x\,(\;x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x, y))\;)$$

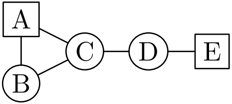


green=true red=false

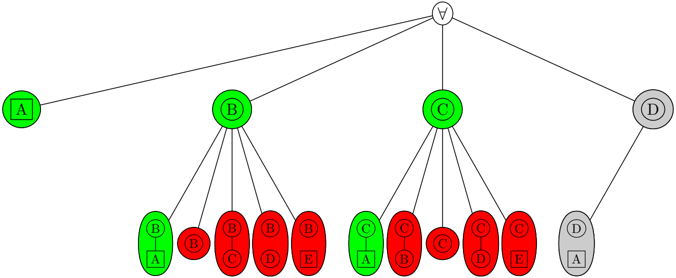


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y \,(y \in U \wedge adj(x,y))\;)$$

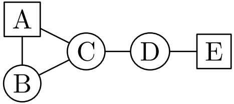


green=true red=false

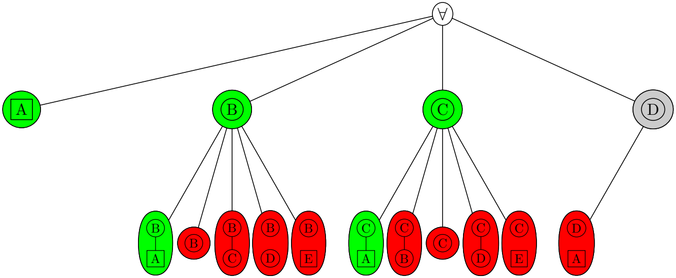


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

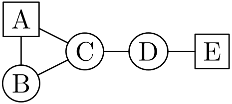


green=true red=false

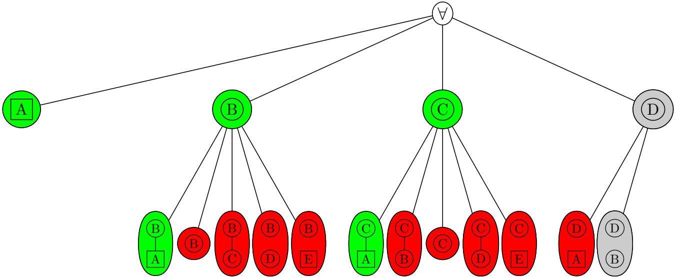


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

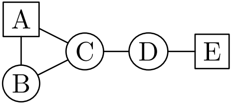


green=true red=false

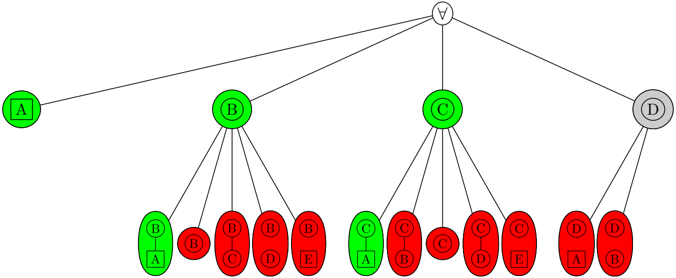


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

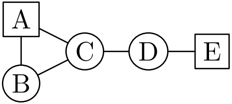


green=true red=false

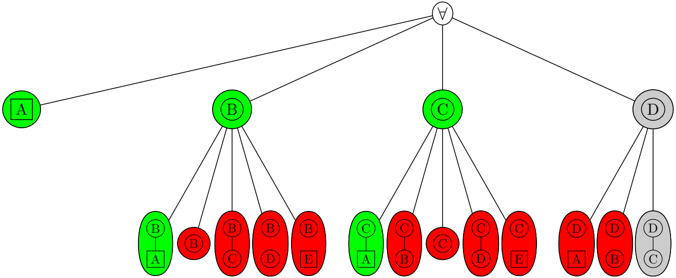


A Game with Two Players

$$ds(U) = \forall x\, (\ x \in U\ \ \vee\ \ \exists y\, (y \in U \wedge adj(x, y))\ )$$

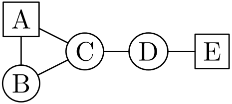


green=true red=false

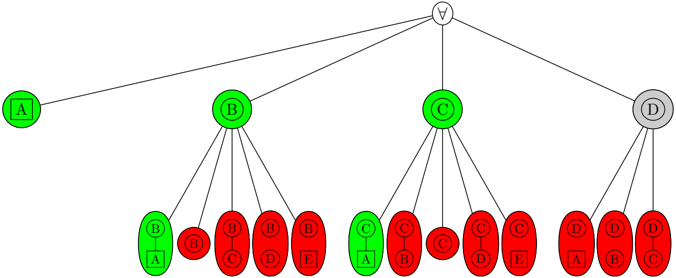


A Game with Two Players

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

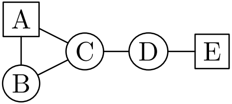


green=true red=false

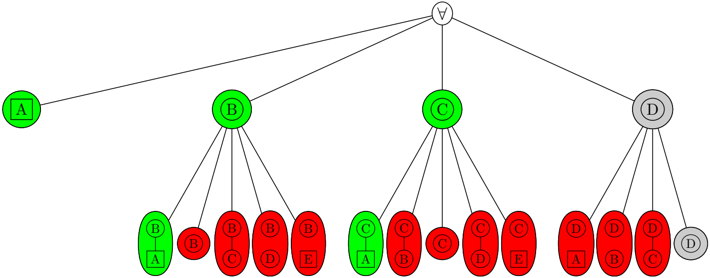


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \ \vee\ \ \exists y\,(y \in U \wedge adj(x,y))\ )$$

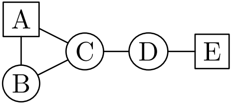


green=true red=false

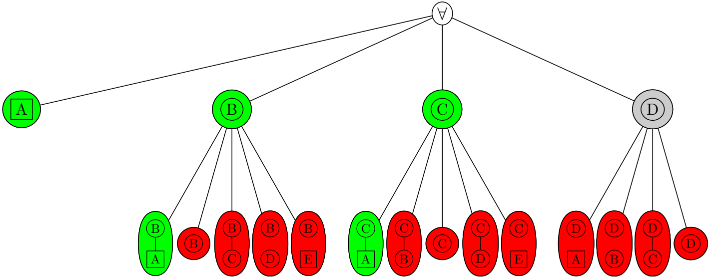


A Game with Two Players

$$ds(U) = \forall x\, (\; x \in U \;\;\vee\;\; \exists y\, (y \in U \wedge adj(x, y))\;)$$

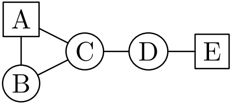


green=true red=false

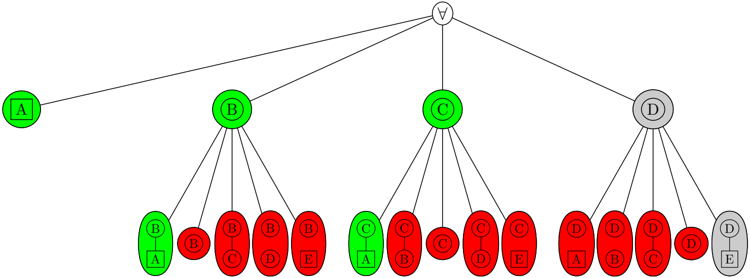


A Game with Two Players

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

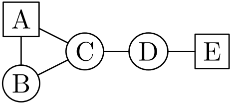


green=true red=false

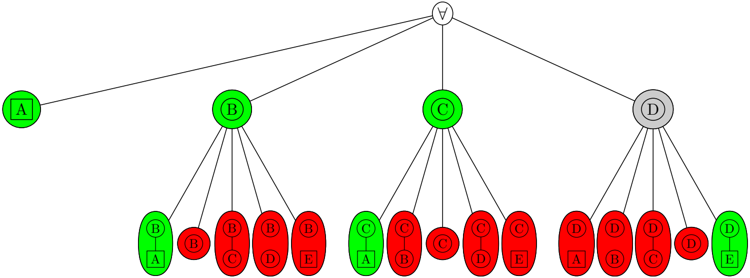


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x, y))\;)$$

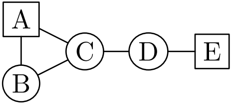


green=true red=false

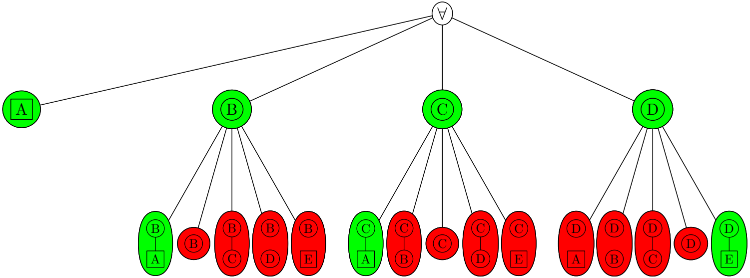


A Game with Two Players

$$ds(U) = \forall x \, (\; x \in U \; \vee \; \exists y \, (y \in U \wedge adj(x,y)) \;)$$

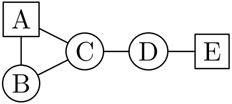


green=true red=false

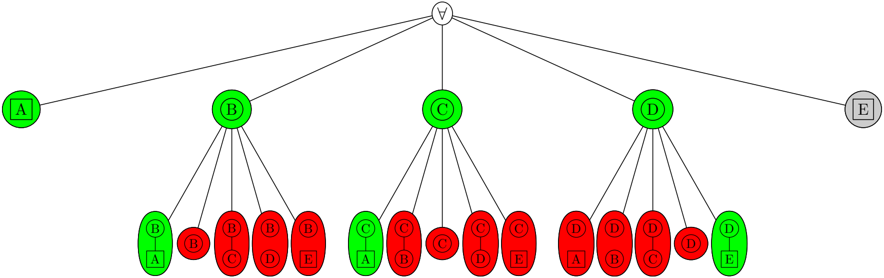


A Game with Two Players

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

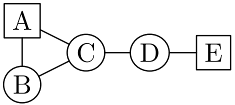


green=true red=false

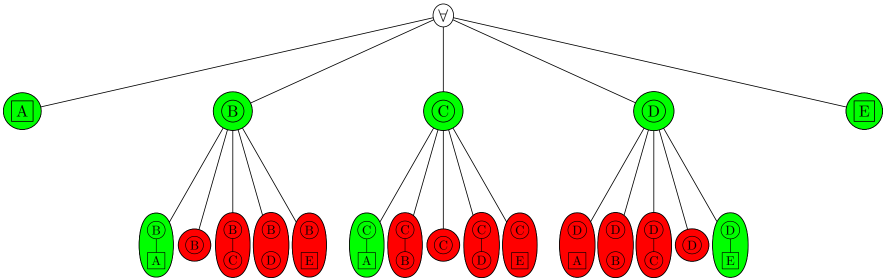


A Game with Two Players

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

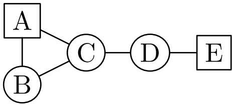


green=true red=false

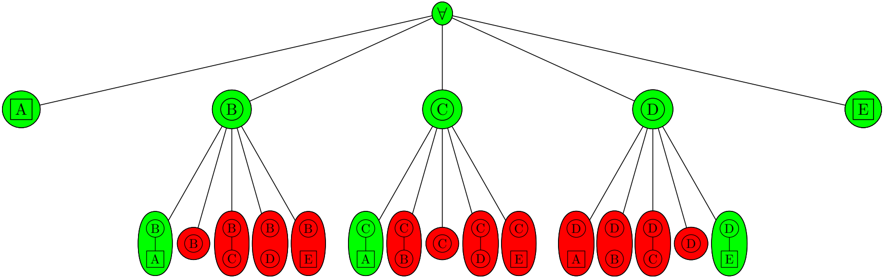


A Game with Two Players

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$

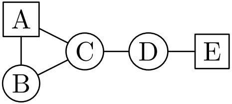


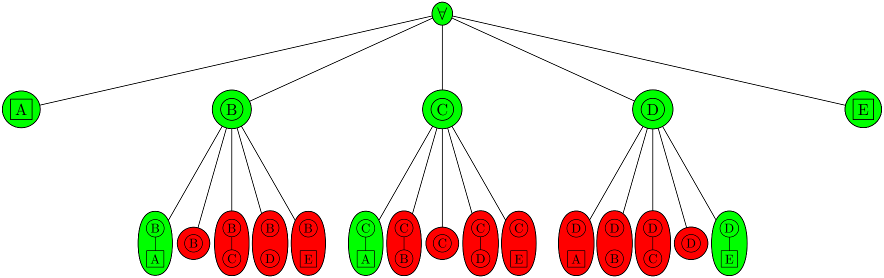
green=true red=false



A Game with Two Players

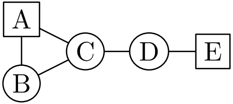
$$ds(U) = \forall x \, (\; x \in U \;\; \vee \;\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$

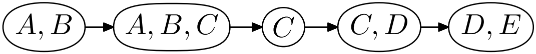




Partial Evaluation + Placeholders

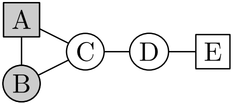
$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

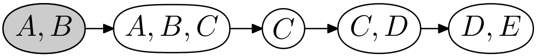




Partial Evaluation + Placeholders

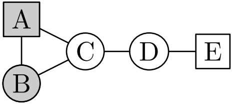
$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y \,(y \in U \wedge adj(x, y))\;)$$

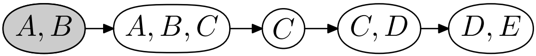




Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

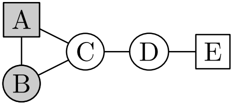


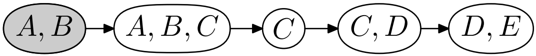


green=true red=false

Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\ x \in U\ \lor\ \exists y\,(y \in U \land adj(x,y))\ )$$



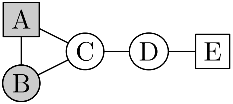


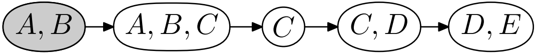
green=true red=false

$\forall$

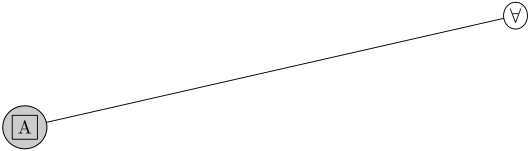
Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\ x \in U\ \ \vee\ \ \exists y\,(y \in U \wedge adj(x,y))\ )$$



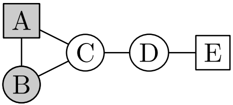


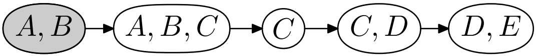
green=true red=false



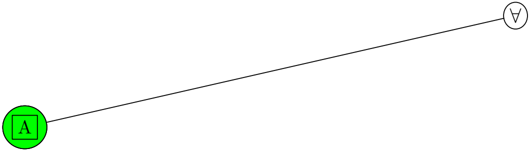
Partial Evaluation + Placeholders

$$ds(U) = \forall x\, (\ x \in U\ \vee\ \exists y\, (y \in U \wedge adj(x,y))\ )$$



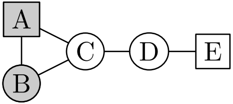


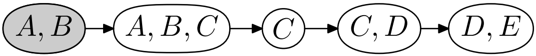
green=true red=false



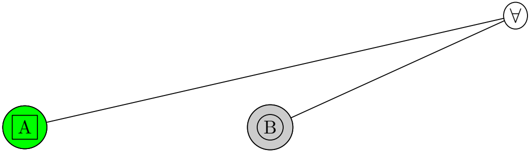
Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\ x \in U\ \ \vee\ \ \exists y\,(y \in U \wedge adj(x,y))\ )$$



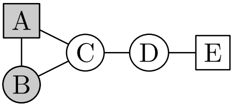


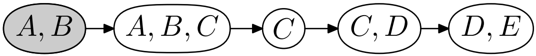
green=true red=false



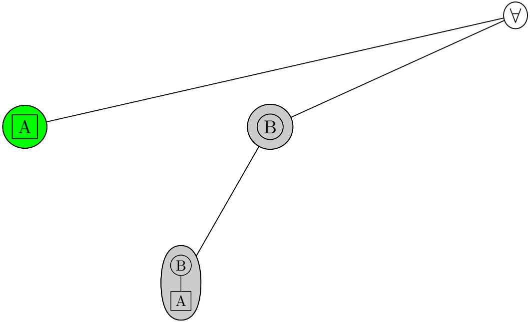
Partial Evaluation + Placeholders

$$ds(U) = \forall x \; (\; x \in U \;\; \vee \;\; \exists y \, (y \in U \wedge adj(x,y)) \;)$$



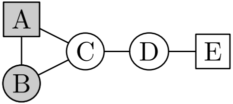


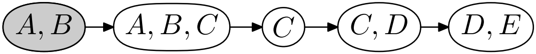
green=true red=false



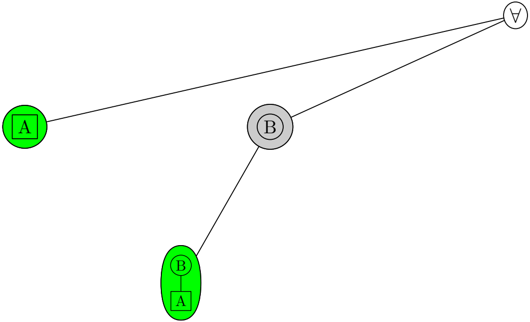
Partial Evaluation + Placeholders

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$



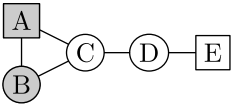


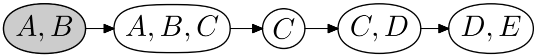
green=true red=false



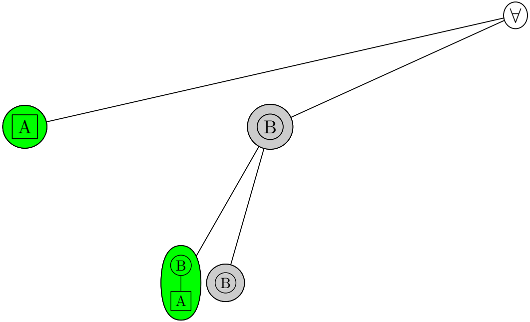
Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$



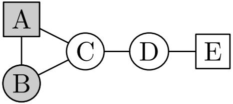


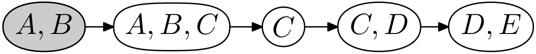
green=true red=false



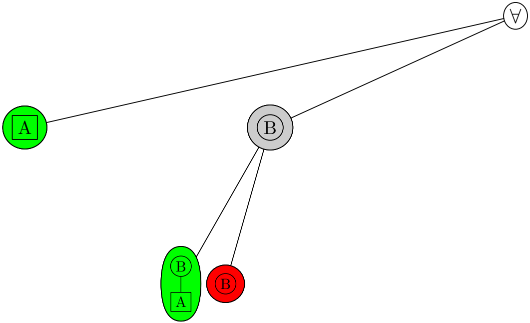
Partial Evaluation + Placeholders

$$ds(U) = \forall x \, (\; x \in U \;\vee\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$



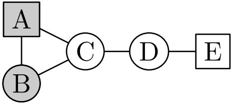


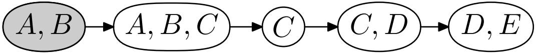
green=true red=false



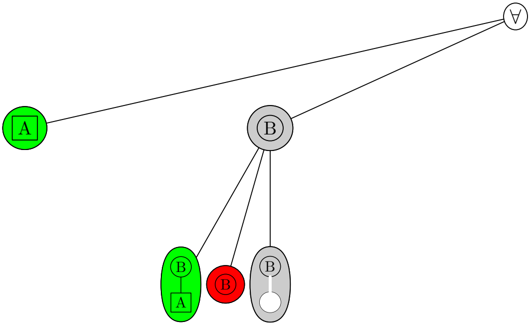
Partial Evaluation + Placeholders

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x, y))\;)$$



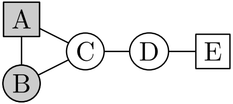


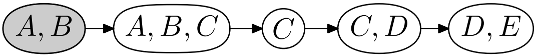
green=true red=false



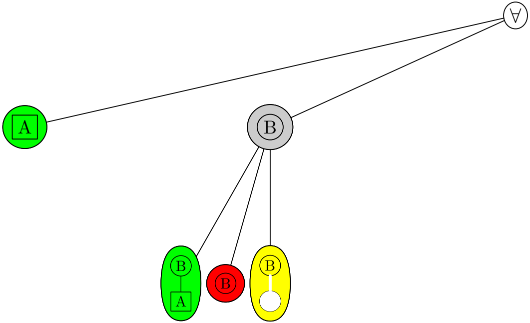
Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$



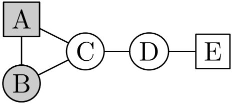


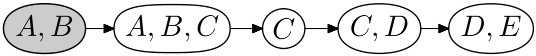
green=true red=false yellow=don't know



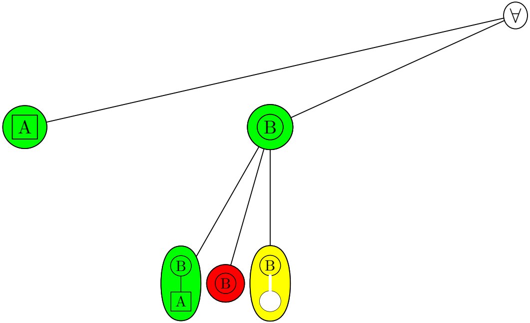
Partial Evaluation + Placeholders

$$ds(U) = \forall x \, (\; x \in U \;\; \vee \;\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$



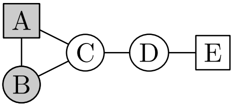


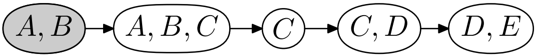
green=true red=false yellow=don't know



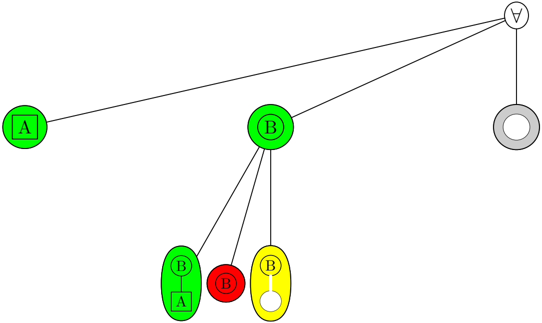
Partial Evaluation + Placeholders

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$



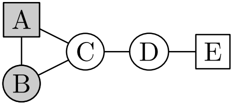


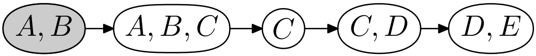
green=true red=false yellow=don't know



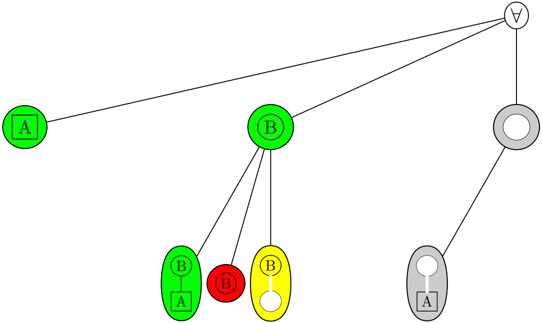
Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$



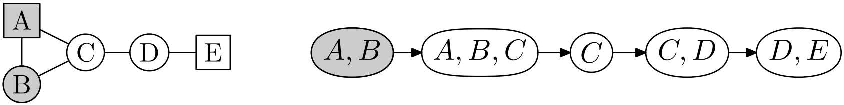


green=true red=false yellow=don't know

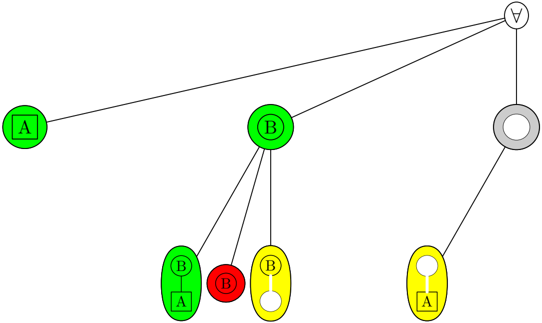


Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$

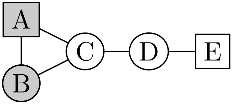


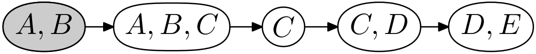
green=true red=false yellow=don't know



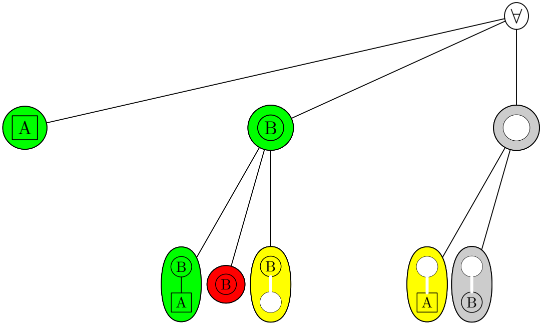
Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$



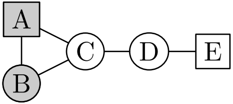


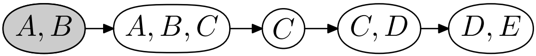
green=true red=false yellow=don't know



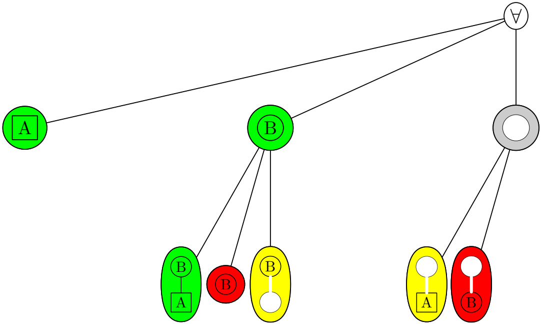
Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$



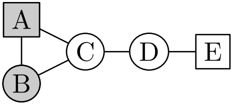


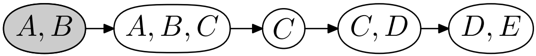
green=true red=false yellow=don't know



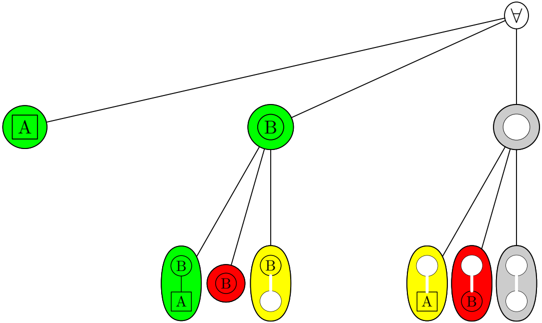
Partial Evaluation + Placeholders

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$



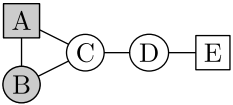


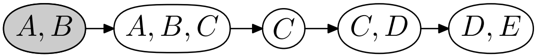
green=true red=false yellow=don't know



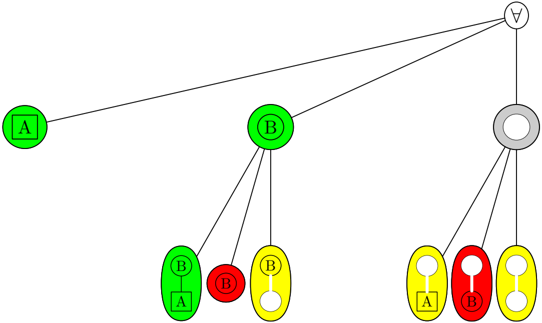
Partial Evaluation + Placeholders

$$ds(U) = \forall x\,(\ x \in U\ \vee\ \exists y\,(y \in U \wedge adj(x,y))\ )$$



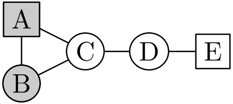


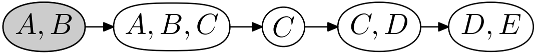
green=true red=false yellow=don't know



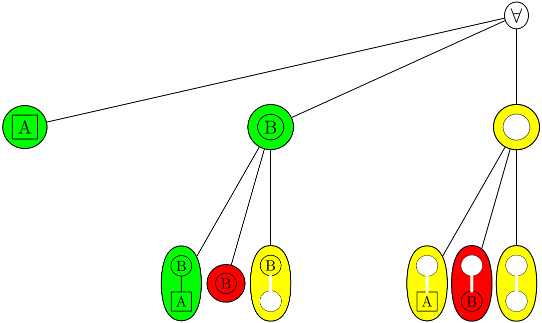
Partial Evaluation + Placeholders

$$ds(U) = \forall x \,(\; x \in U \;\vee\; \exists y\,(y \in U \wedge adj(x,y))\;)$$



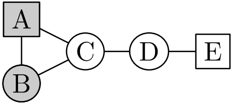


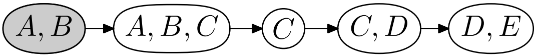
green=true red=false yellow=don't know



Partial Evaluation + Placeholders

$$ds(U) = \forall x \, (\; x \in U \;\; \vee \;\; \exists y \, (y \in U \wedge adj(x, y)) \;)$$





green=true red=false yellow=don't know