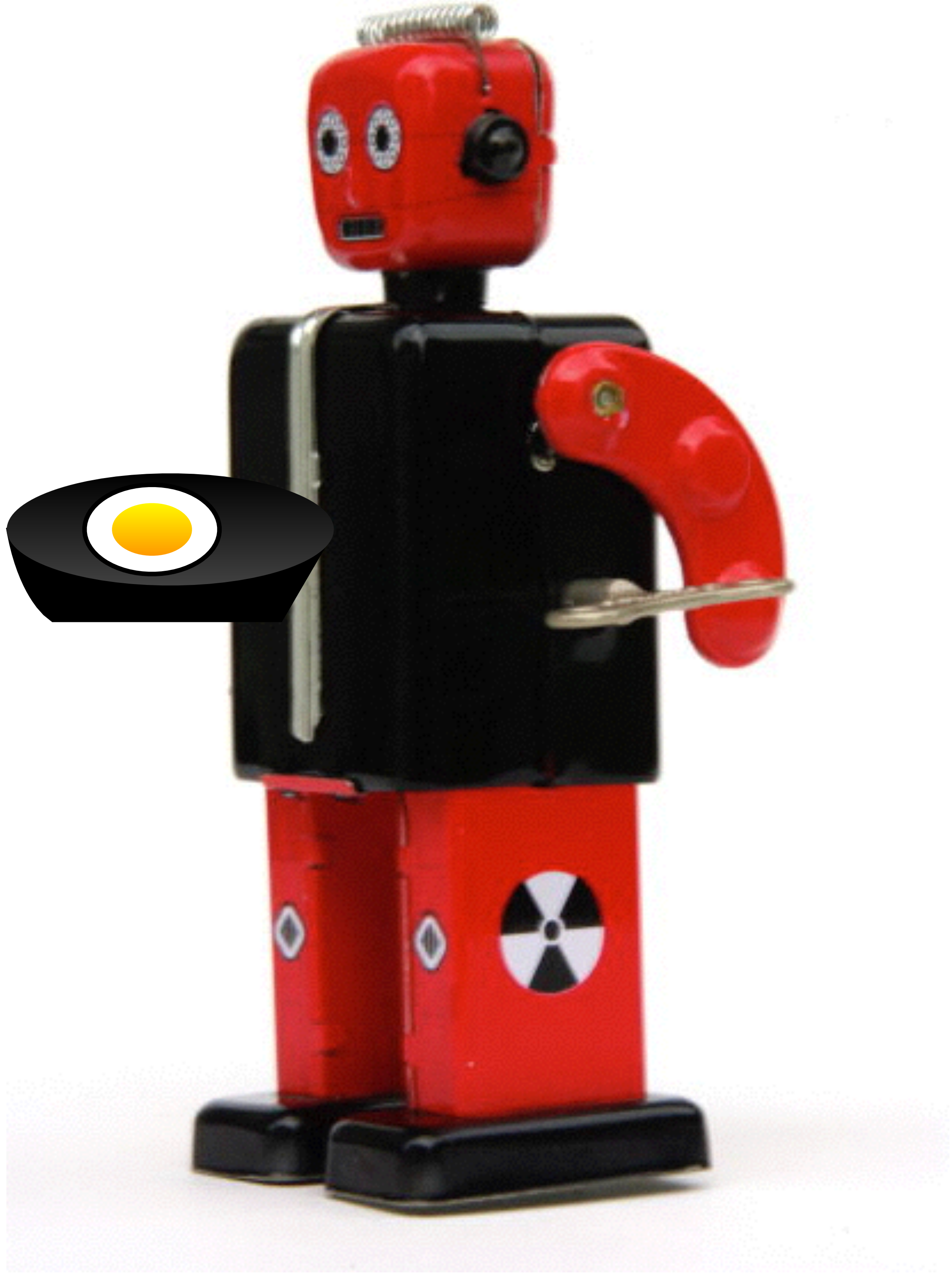
Topological Sorting

Andreas Klappenecker

Breakfast Robot



Example

There are some tasks that need to be done to eat breakfast:

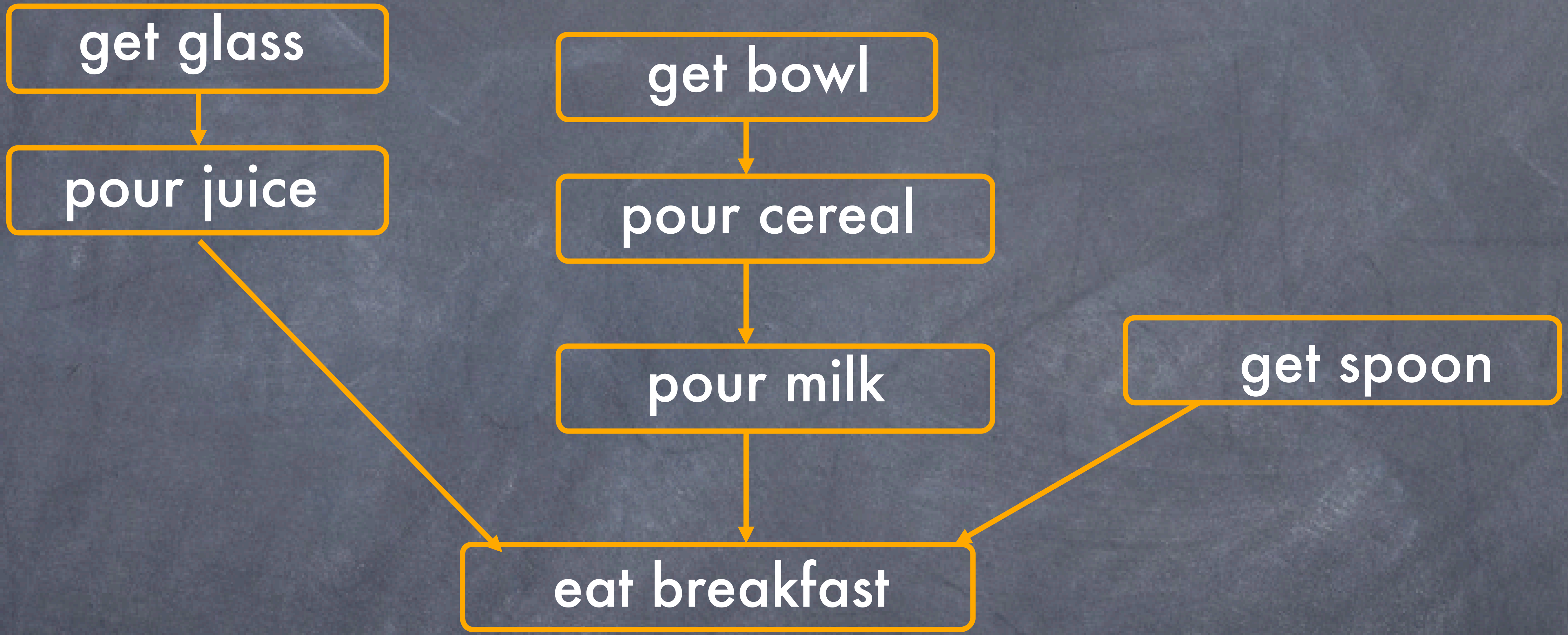
get glass, pour juice, get bowl, pour cereal,

pour milk, get spoon, eat.

Some of the events must take precedence over others. For example, “get bowl” should precede “pour milk”.

The ordering of some other events is irrelevant, e.g., “get g/b/s”.

Example



Goal: Embed the partial order of events into a total order

Partial Order Relation

Let S be a set and $\preccurlyeq$ a relation on S. Then $\preccurlyeq$ is called a partial order if and only if for all a, b, c in S, we have

- $a \preccurlyeq a$ (reflexivity)
- if $a \preccurlyeq b$ and $b \preccurlyeq a$, then $a = b$ (antisymmetry)
- if $a \preccurlyeq b$ and $b \preccurlyeq c$, then $a \preccurlyeq c$ (transitivity)

Any partial order can be embedded into a total order.

Cover Relation

Let $\preccurlyeq$ be a partial order on a set S. The cover relation $\lessdot$ of this partial order is defined as

$a \lessdot b$ if and only if $a \preccurlyeq b$ and there doesn't exist x s.t. $a < x < b$.

Two elements are related under the cover relation iff their are immediate neighbors in the partial order.

Representation

Let $(S, \leqslant)$ be a partial order, and $\lessdot$ its cover relation.

Let R be any relation on S such that

- $\lessdot$ is contained in R,
- R is contained in $\leqslant$

Then the reflexive and transitive closure of R is $\leqslant$.

The relation R can be represented by a directed acyclic graph.

Topological Sorting

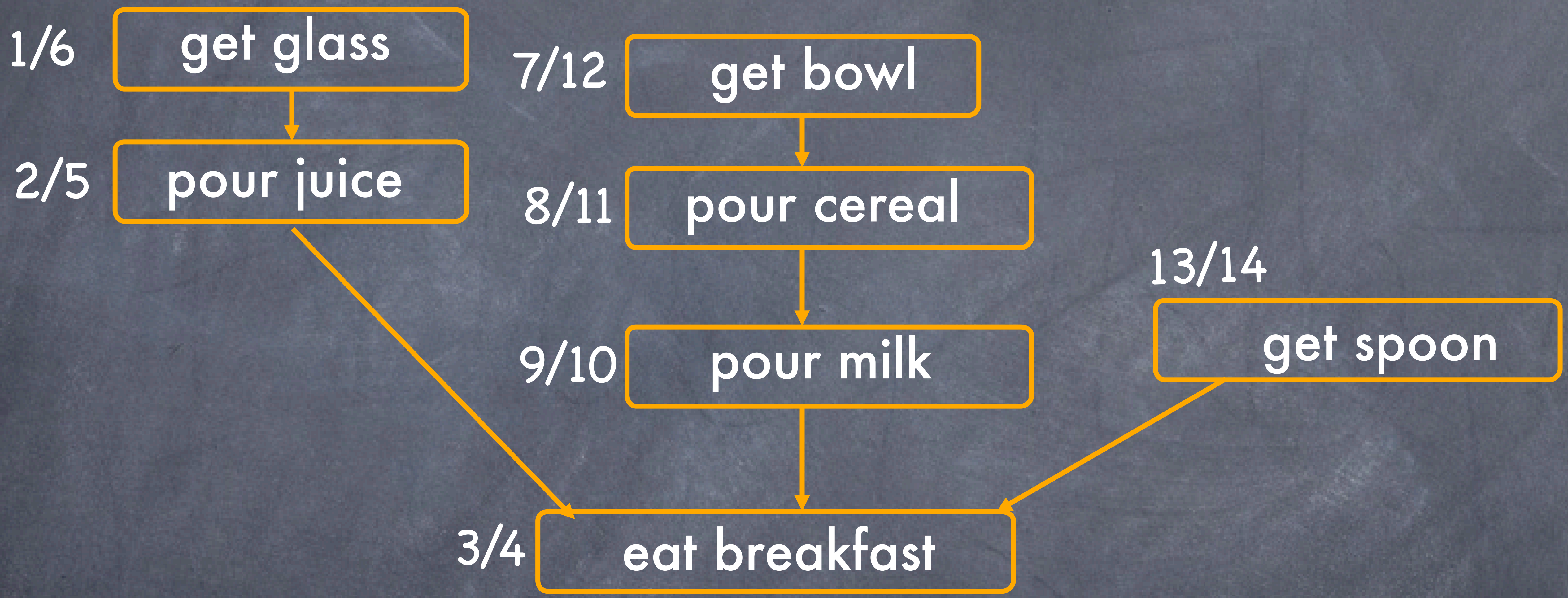
Let G=(S,E) be a directed acyclic graph.

Then G represents a partial order.

Goal: Find a total order $\leq$ on S such that if (u,v) in E, then $u \leq v$.

This can be solved, since any partial order can be embedded into a total order.

Example



get spoon ≤ get bowl ≤ pour cereal ≤ pour milk
≤ get glass ≤ pour juice ≤ eat breakfast

Topological Sorting Algorithm

Input: Directed acyclic graph G = (V,E)

1. Call DFS on G to compute finish[v] for all nodes v
2. After a node's recursive call finishes, insert it at the front of a linked list
3. return the linked list (so, events are ordered by decreasing finishing time).

Running Time: O(V+E)

Correctness

Let e = (u,v) be an edge of the directed acyclic graph G=(V,E).

- If e is a forward or tree edge, then finish[v] < finish[u].
- If e is a cross edge, then finish[v] < disc[u] < finish[u].
- The edge e cannot be a back edge, since G is acyclic.

Therefore, finish[u] > finish[v] in all cases. Thus, the total order produced by DFS respects the partial order implied by G.

Credits

Many thanks to Jennifer Welch for providing the breakfast example and breakfast priority graph.