ON BAYESIAN NETWORK INFERENCE WITH SIMPLE PROPAGATION

Cory J. Butz
butz@cs.uregina.ca
University of Regina
Canada

Jhonatan S. Oliveira
oliveira@cs.uregina.ca
University of Regina
Canada

André E. dos Santos
dossantos@cs.uregina.ca
University of Regina
Canada

Anders L. Madsen
anders@hugin.com
HUGIN EXPERT A/S
Aalborg University
Denmark

OUTLINE

- Bayesian networks
- Inference with Lazy Propagation
- Inference with Simple Propagation
- Experimental Results & Analysis
- Conclusions

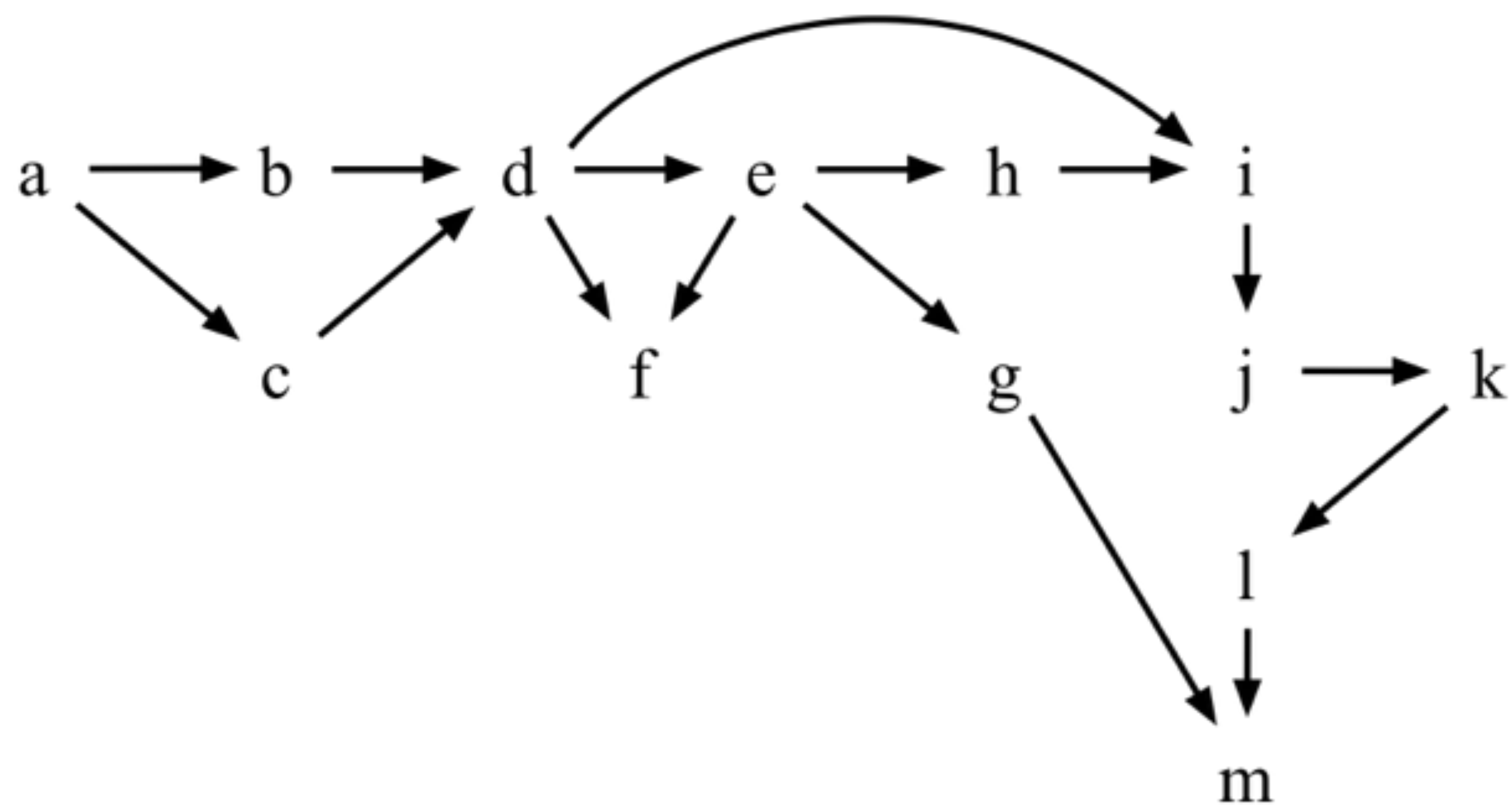
Bayesian Networks

A Bayesian Network (BN) consists of:

- a directed acyclic graph (DAG)
- a matching set of conditional probability tables (CPTs)

The product of the CPTs is a join probability distribution (JPD) *P(U)*

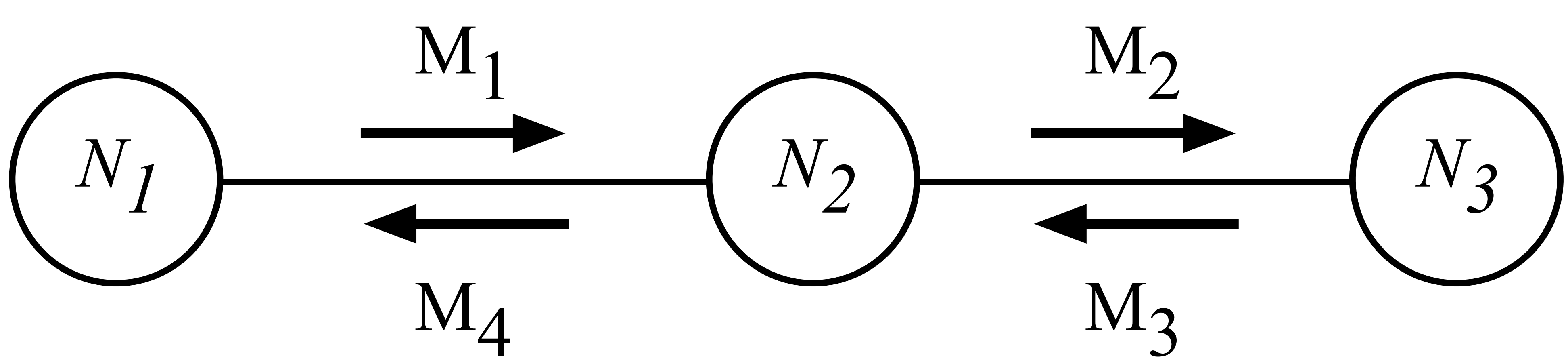
Bayesian Network Example



$$P(U) = P(a) \bullet P(b|a) \bullet P(c|a) \bullet P(d|b,c) \bullet \ldots \bullet P(m|g,l)$$

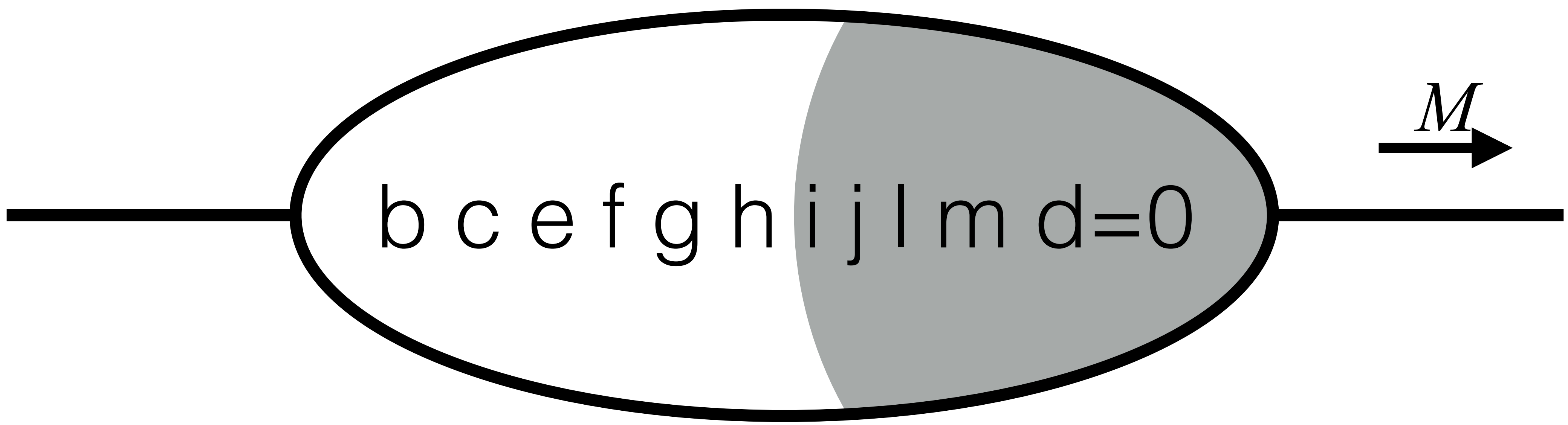
LAZY PROPAGATION

- Madsen and Jensen (AIJ 1999)
- BN variables are clustered into nodes
- Nodes are organized as a join tree
- Each BN CPT is assigned to a join tree node
- Messages are propagated systematically



Message Construction

$$message \ = \sum_{N-N'} Factorization \ at \ N$$



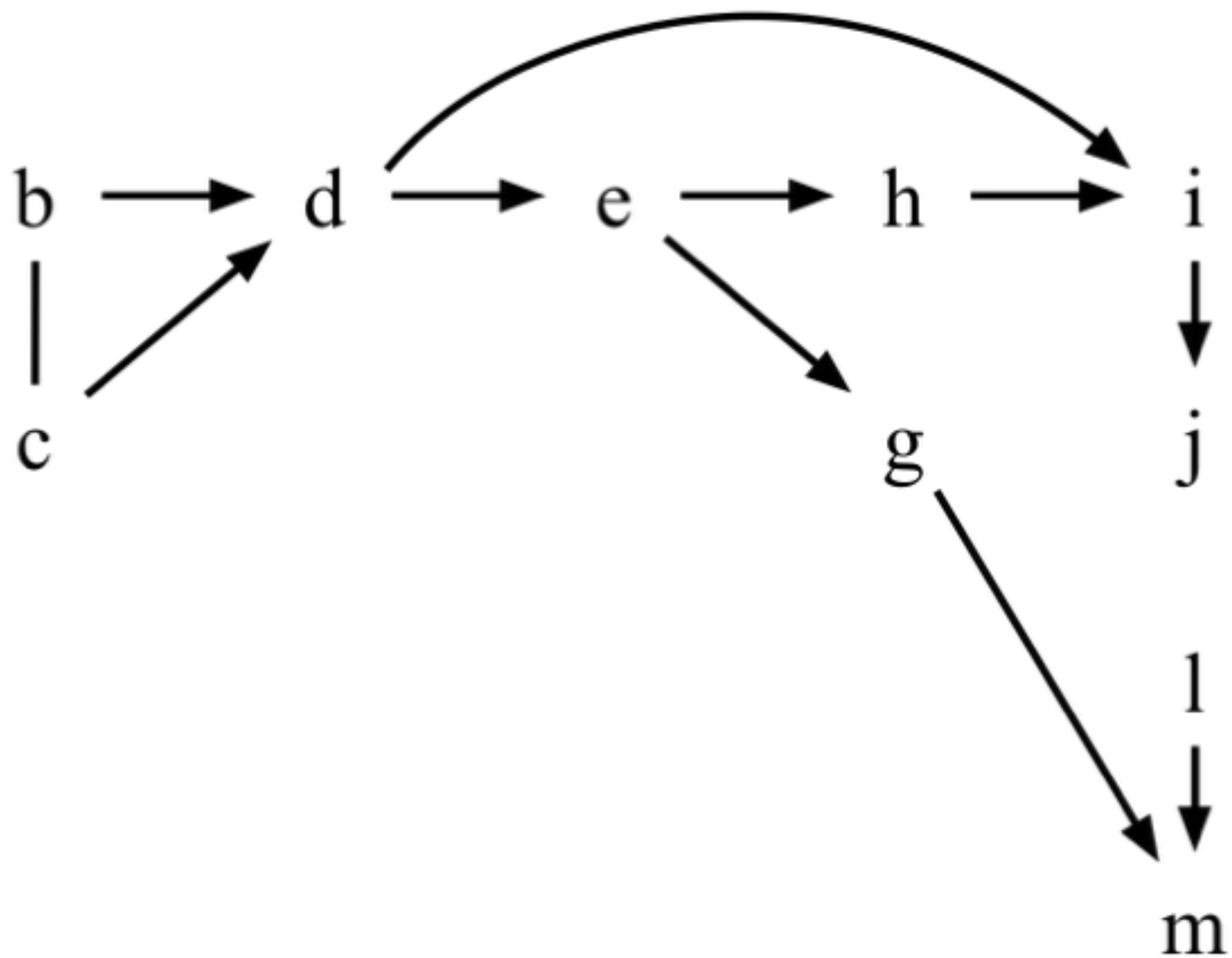
$$M \ = \sum_{b,c,e,f,g,h} P(b,c) \cdot P(d=0|b,c) \cdot P(e|d=0) \cdot P(f|d=0,e)$$

$$\cdot P(g|e) \cdot P(h|e) \cdot P(i|d=0,h) \cdot P(j|i) \cdot P(m|g,l)$$

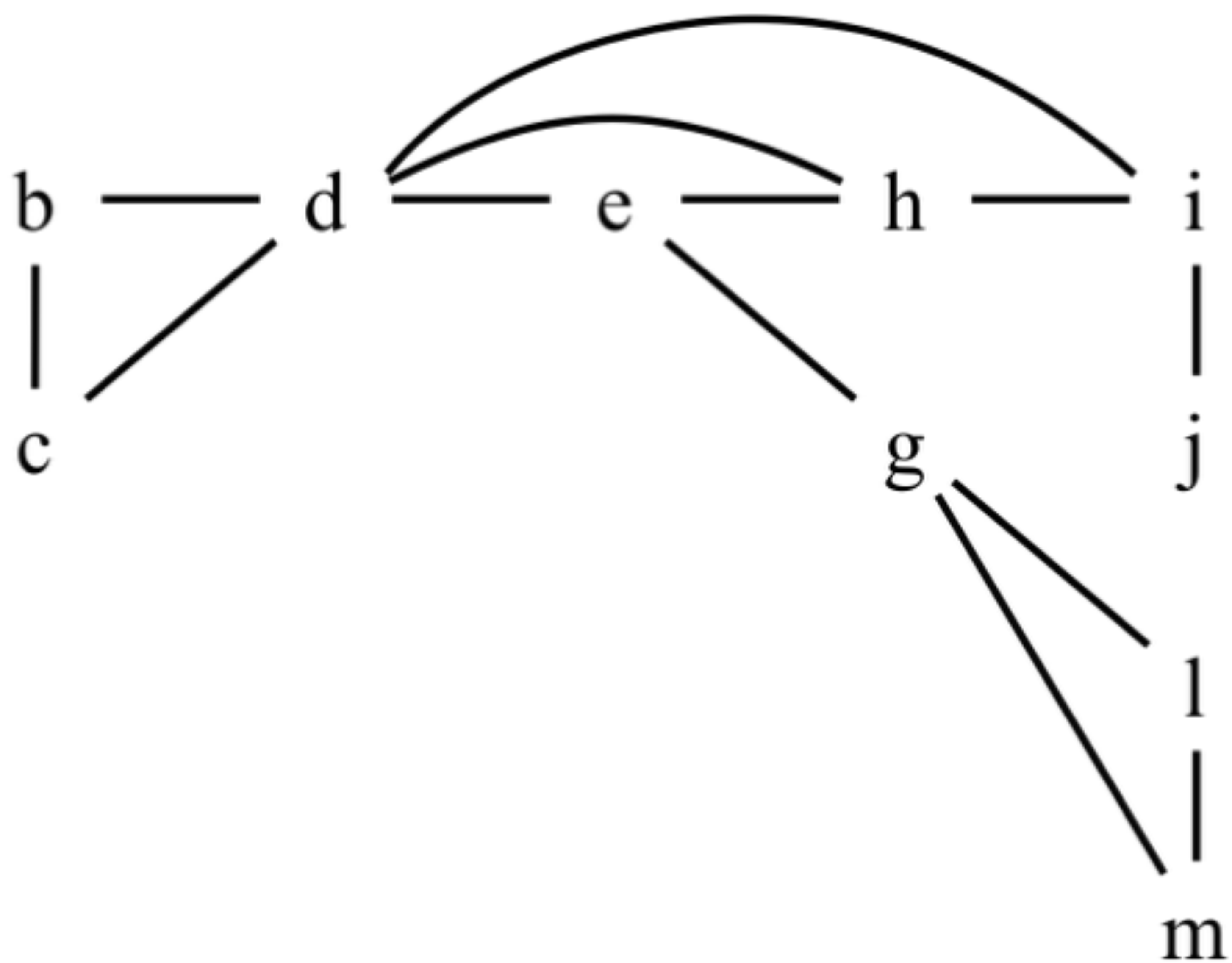
DETECTING IRRELEVANT POTENTIALS

- LP constructs:
  - the domain graph G_1 of the factorization
  - the moralization G_1^m of G_1
- LP tests whether the evidence separates the variables to be marginalized from the separator
- if separated, the potential is irrelevant

BUILD DOMAIN GRAPH G_1

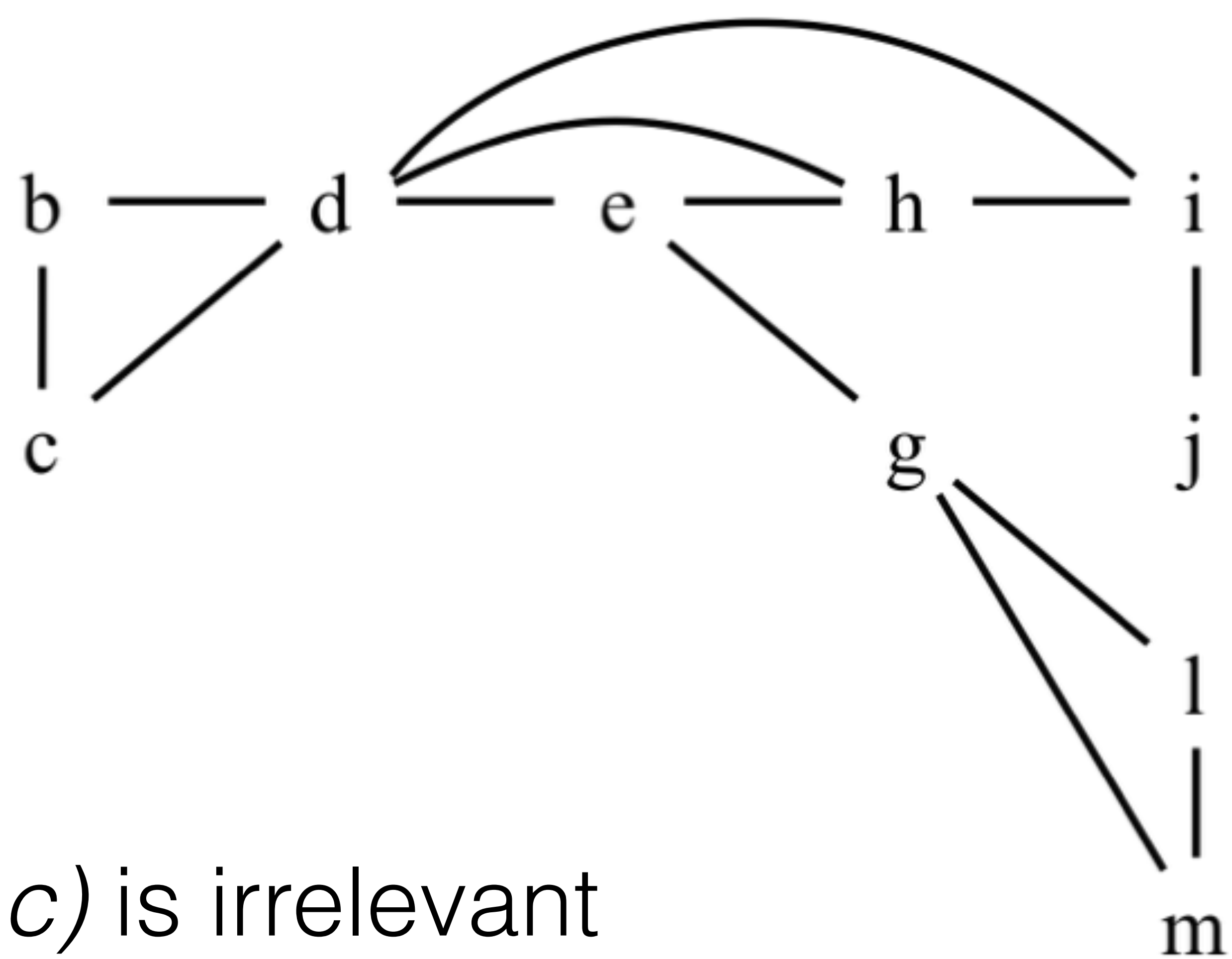


BUILD MORALIZATION GRAPH G_1^m



TEST INDEPENDENCE FOR EACH POTENTIAL

- For *P(b,c)*, test whether evidence *d* separates *b* and *c* from the separator *S = {i,j,l,m}*

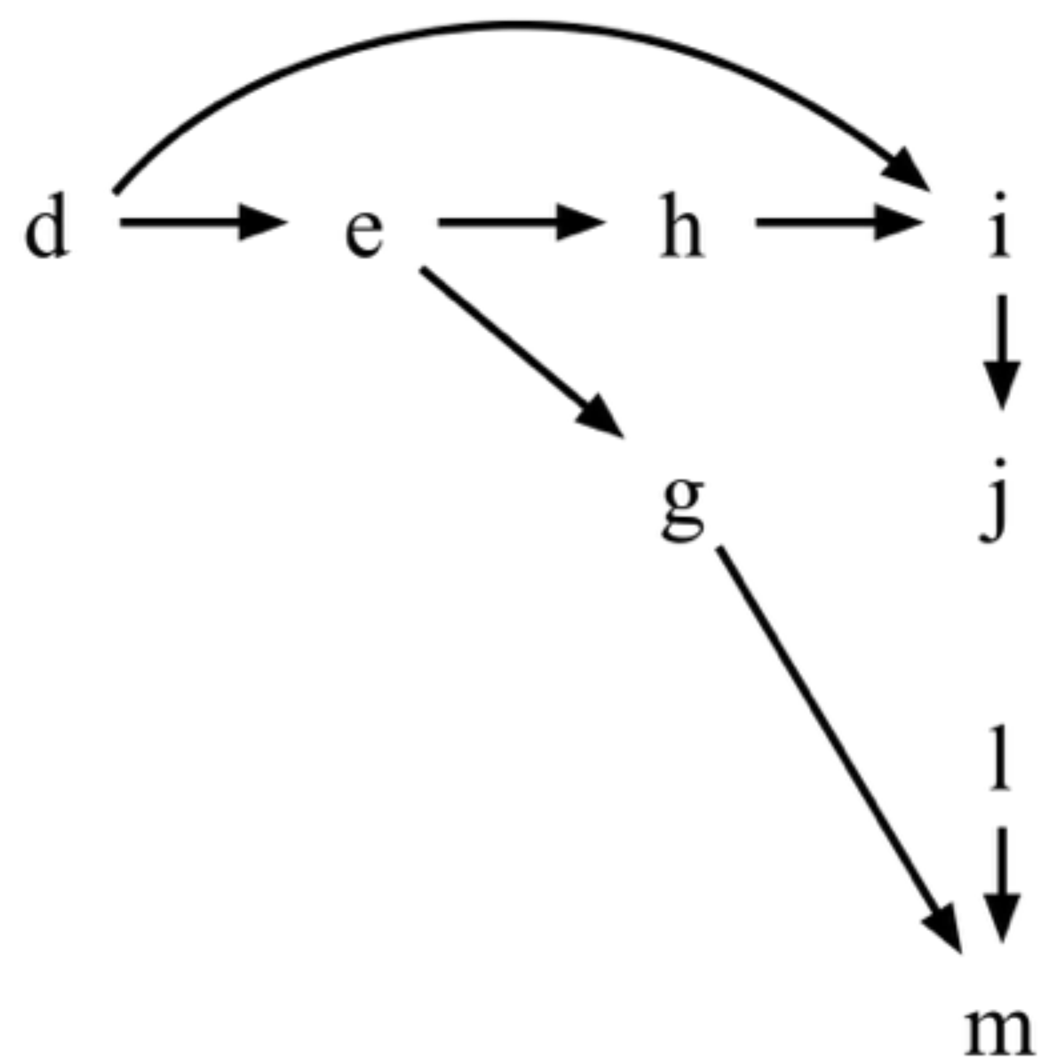


- Thus, *P(b,c)* is irrelevant

Determining Elimination Orderings

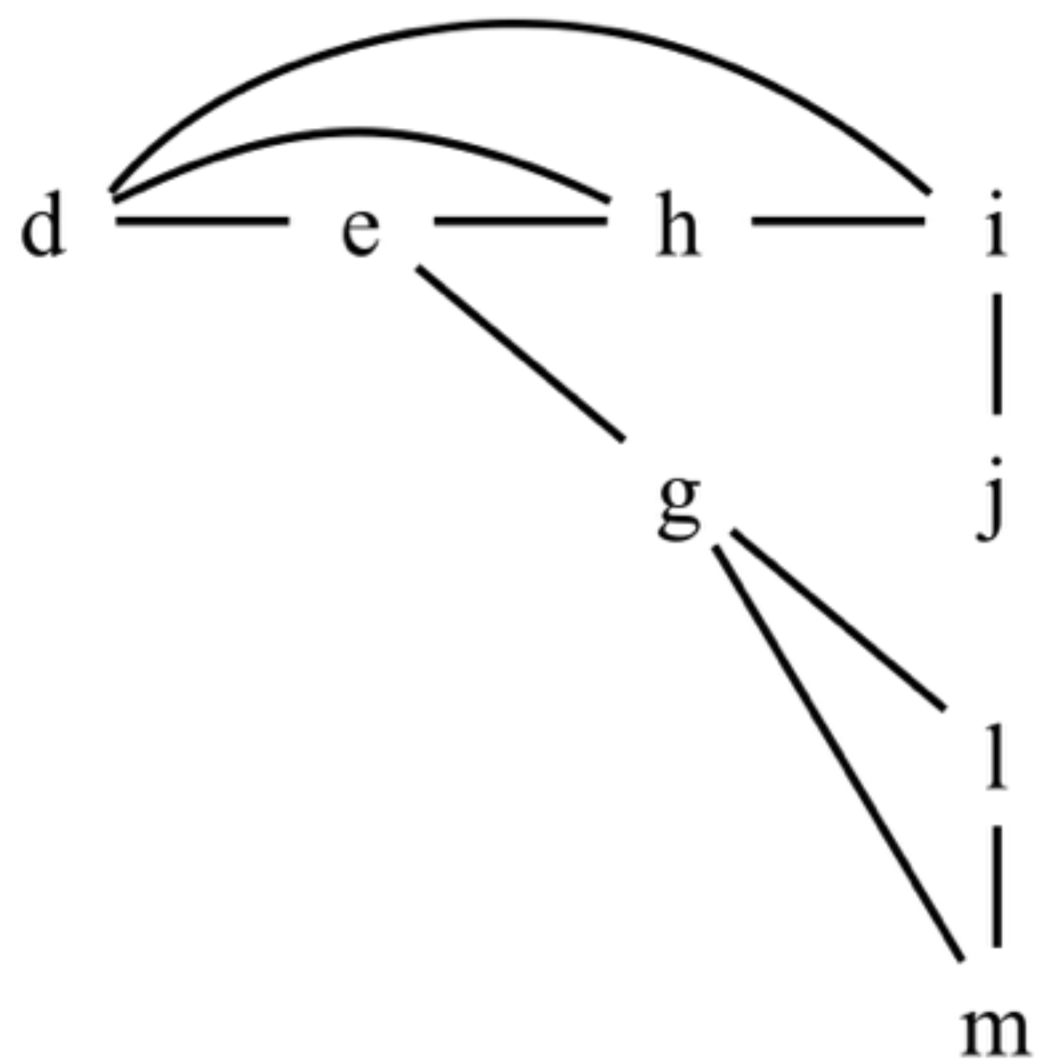
- LP constructs:
  - the domain graph G_2 of the relevant potentials
  - the moralization G_2^m of G_2
- obtain an elimination ordering from G_2^m

BUILD DOMAIN GRAPH G_2

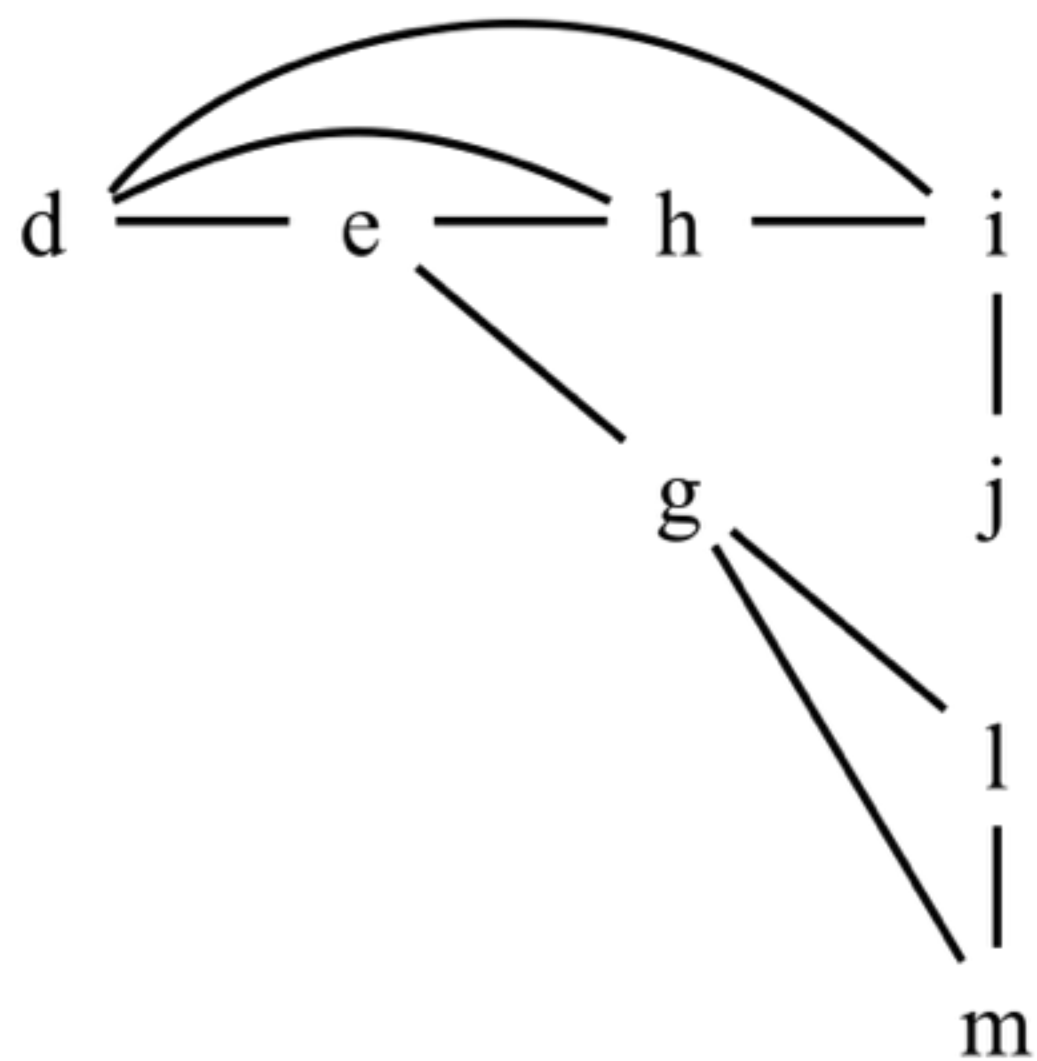


$$\mathcal{F} \ = \ \{P(e|d=0), P(g|e), P(h|e), P(i|d=0,h), P(j|i), P(m|g,l)\}$$

BUILD MORALIZATION GRAPH G_2^m



FIND ELIMINATION ORDERING



- elimination ordering: *g*, *e*, *h*

NOW LP CAN BUILD THE MESSAGE

$$\begin{aligned} M &= \sum_{e,g,h} P(e|d=0)\cdot P(g|e)\cdot P(h|e)\cdot P(i|d=0,h)\cdot P(j|i)\cdot P(m|g,l) \\ &= P(j|i)\cdot \sum_{h} P(i|d=0,h)\cdot \sum_{e} P(e|d=0)\cdot P(h|e)\cdot \sum_{g} P(g|e)\cdot P(m|g,l) \\ &= P(j|i)\cdot P(i,m|d=0,l) \qquad (1) \end{aligned}$$

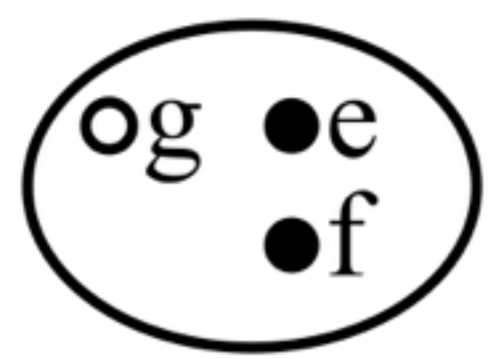
SIMPLE PROPAGATION

DARWINIAN NETWORKS

- Simple Propagation arose from our work on Darwinian Networks (AI 2015)

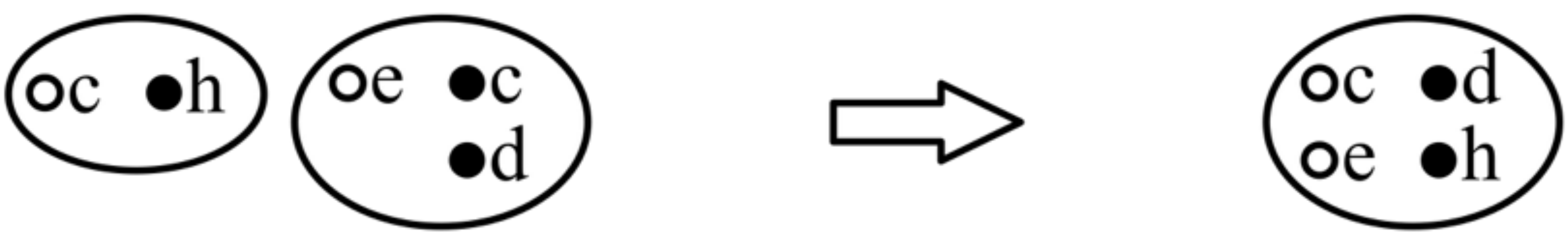
- clever way to view CPTS

$$P(g|e, f) \quad \Rightarrow$$



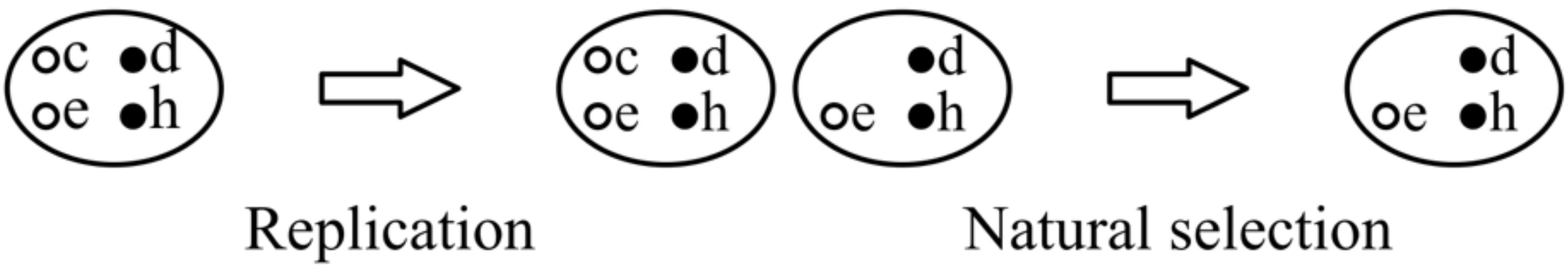
MULTIPLICATION IS MERGE

○ white	+	● black	=	○ white
● black	+	○ white	=	○ white
● black	+	● black	=	● black
○ white	+	○ white	=	● black



$$P(c|h) \cdot P(e|c,d) \quad = \quad P(c,e|d,h)$$

Marginalization is Replication and Natural Selection

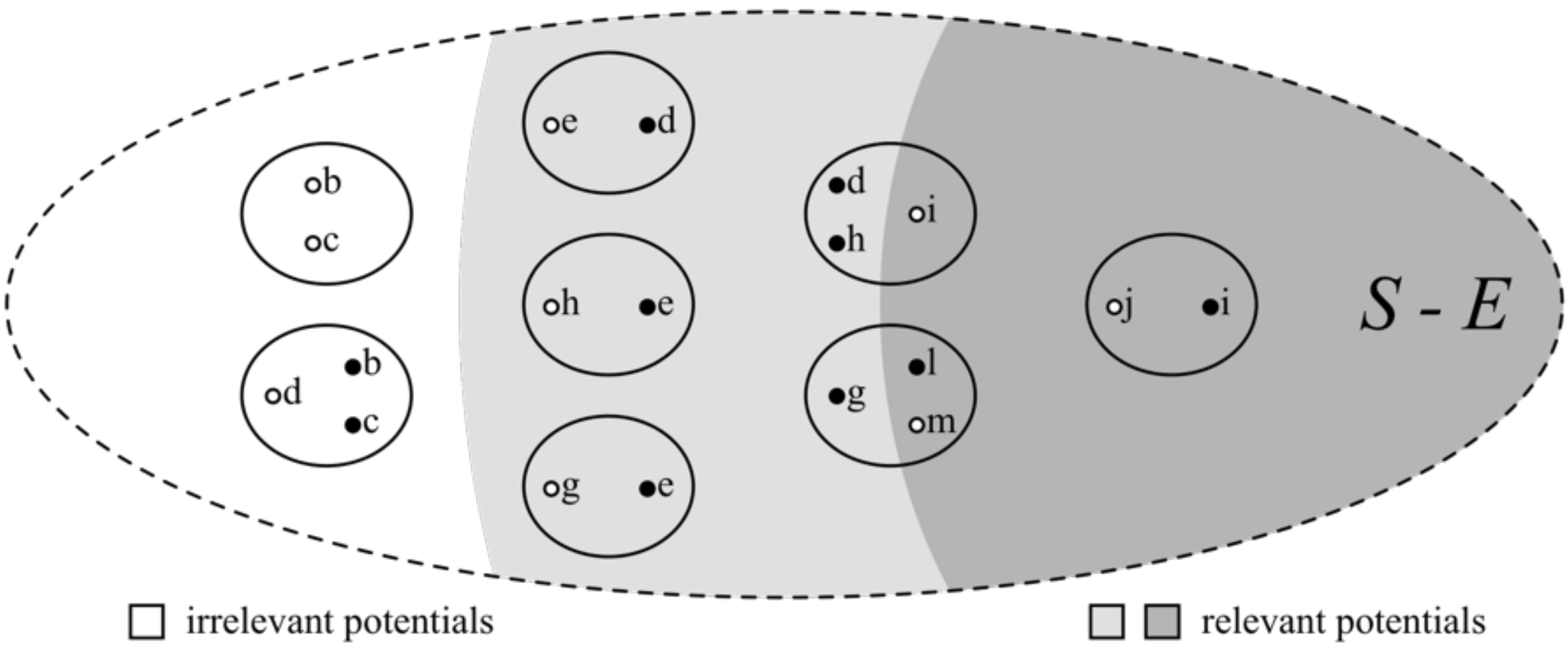


$$\sum_{c} P(c, e|d, h) = P(e|d, h)$$

Simple Propagation

SP only uses the **"one in, one out"** property:

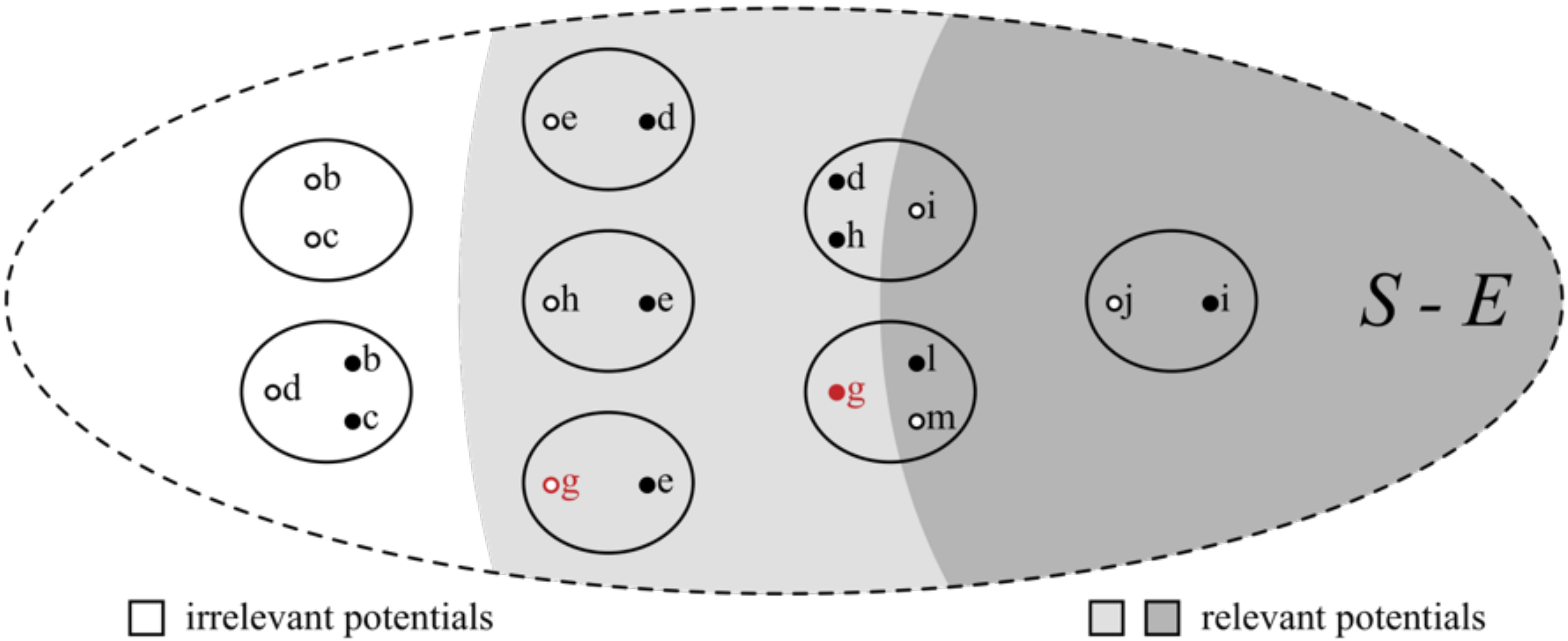
a potential with one non-evidence variable in the separator and another not in the separator



SIMPLE PROPAGATION

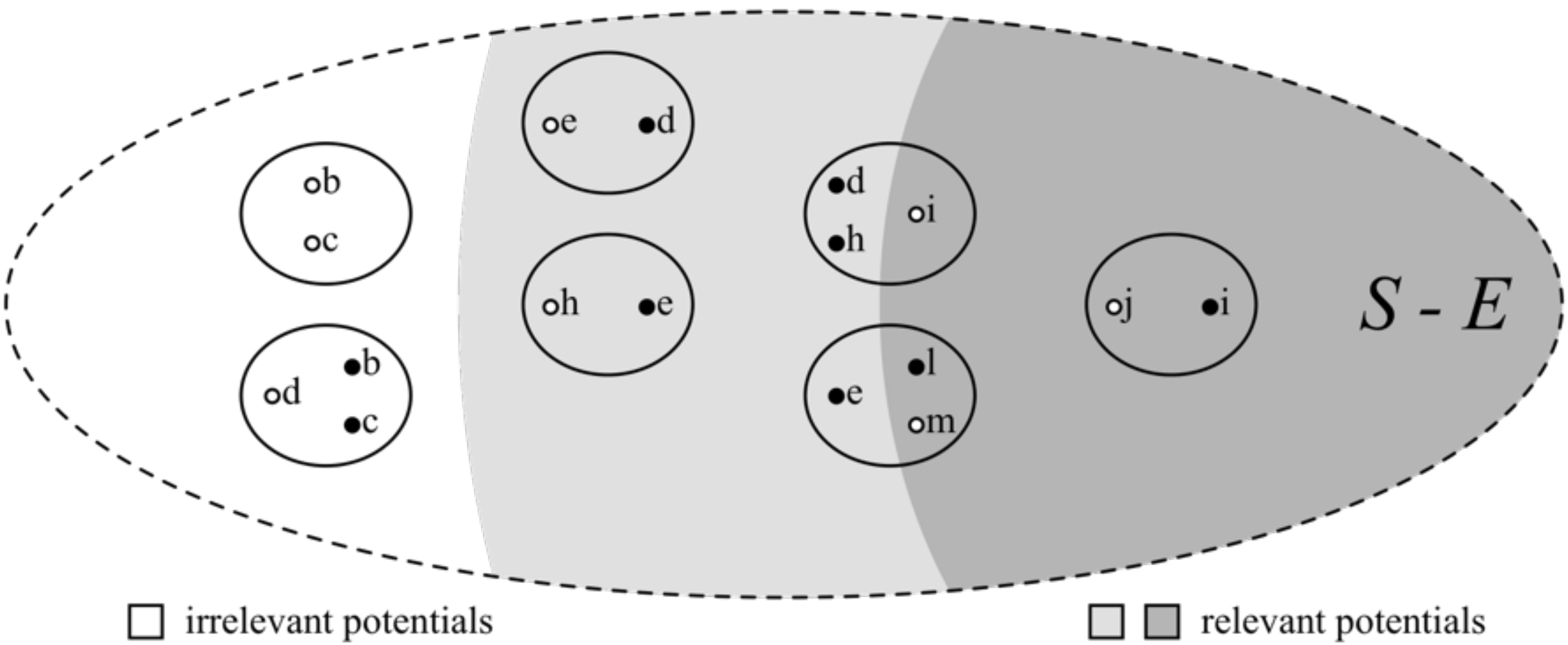
Evidence is $d = 0$

Variable $\boldsymbol{g}$ is outside of *S* and variables *l* and *m* are in *S*



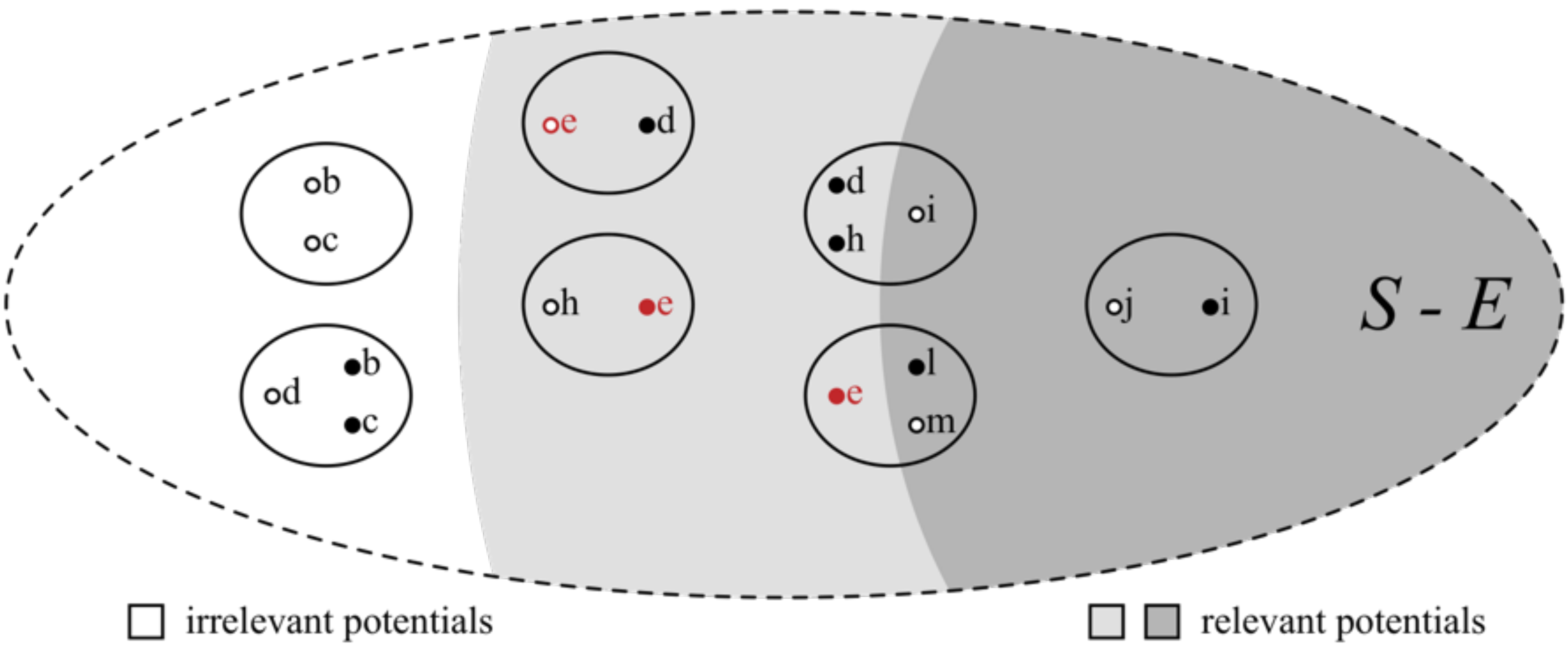
Simple Propagation

Eliminating variable g yields population $p(m|e, l)$



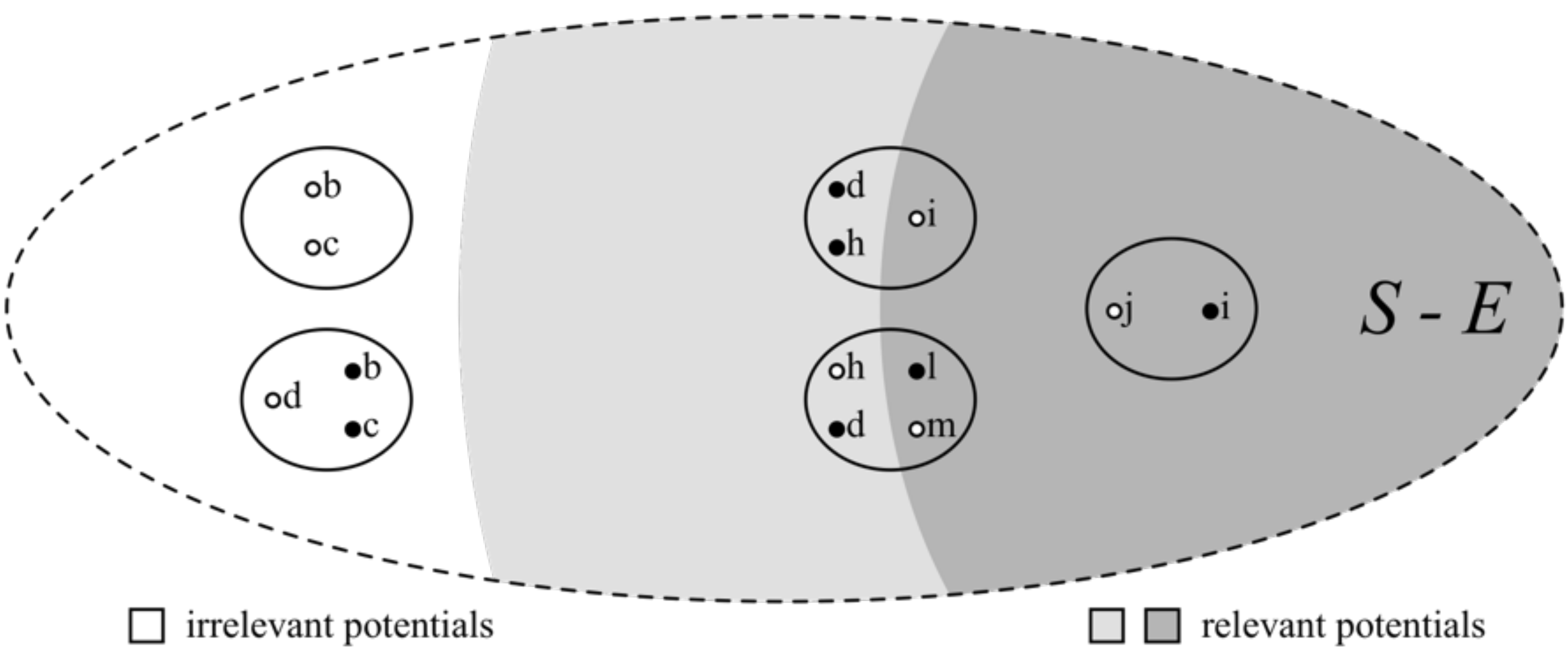
SIMPLE PROPAGATION

Now, variable ***e*** is **out**



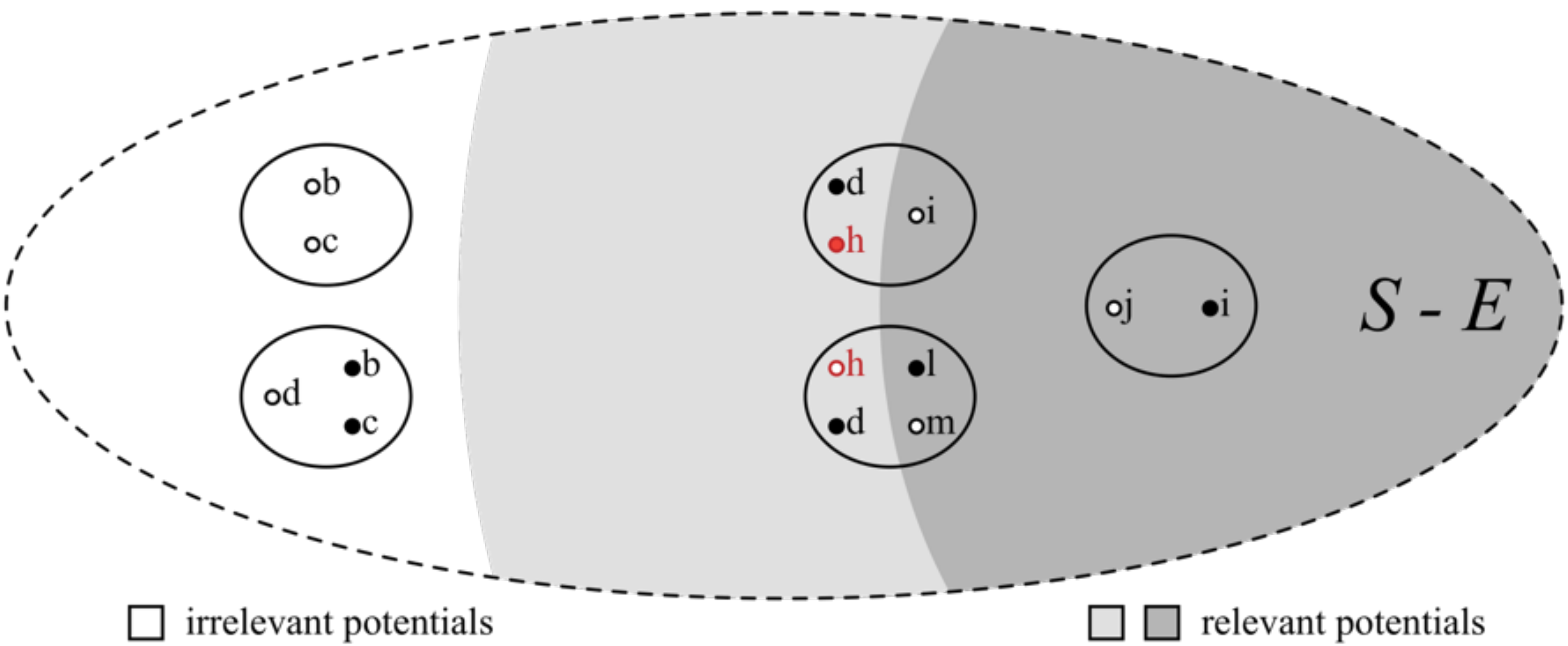
Simple Propagation

Eliminating variable *e* yields $p(h, m \mid d = 0, l)$



SIMPLE PROPAGATION

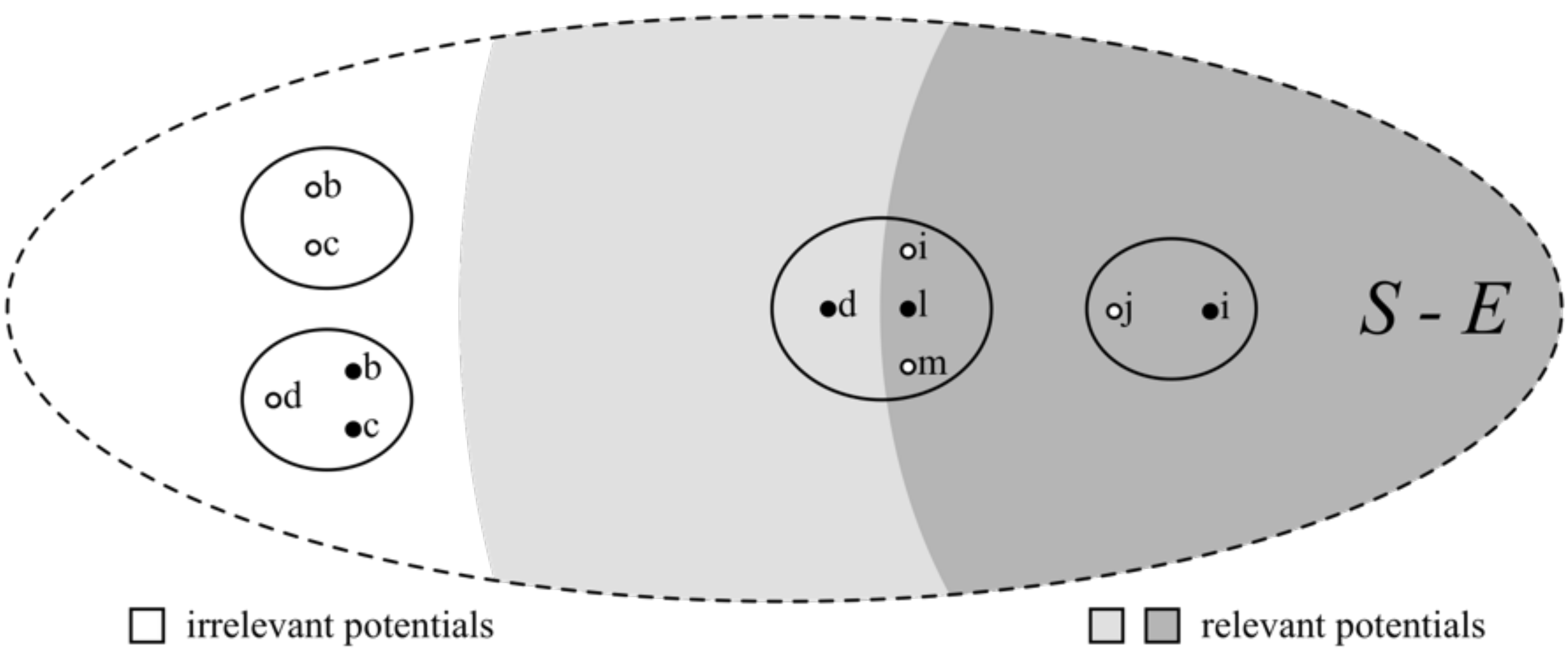
Finally, variable ***h*** is **out**



Simple Propagation

Eliminating variable *h* yields population *p(i, m|d = 0, l)*

$$P(j|i) \cdot P(i, m|d = 0, l) \qquad (1)$$

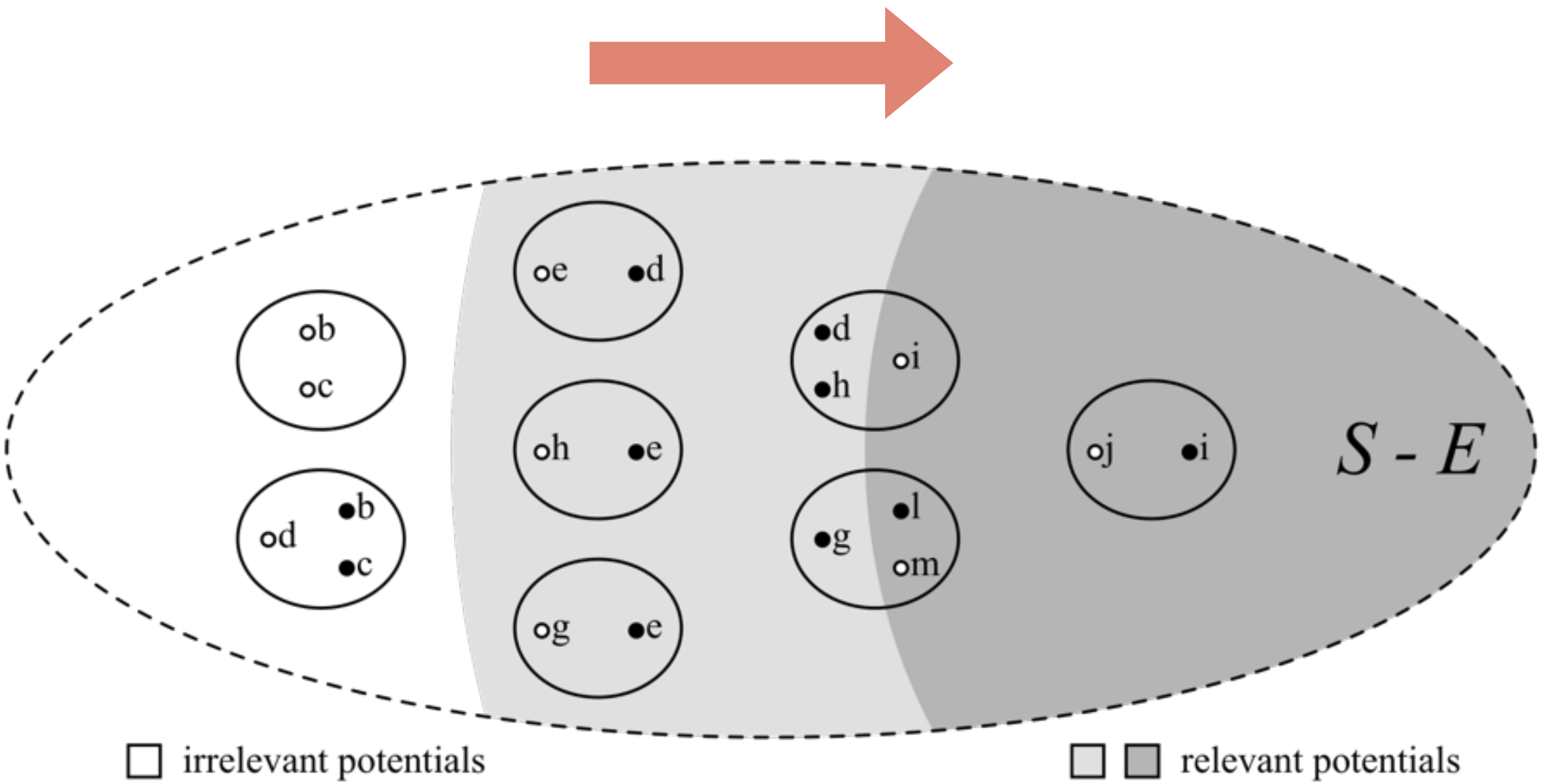


BN	Vars	LP	SP	Saving
Water	32	0.06	0.05	17%
Oow	33	0.07	0.06	14%
Oow_Bas	33	0.04	0.03	25%
Mildew	35	0.05	0.04	20%
Oow_Solo	40	0.07	0.06	14%
Hkv2005	44	0.23	0.27	-17%
Barley	48	0.09	0.1	-11%
Kk	50	0.09	0.09	0%
Ship	50	0.16	0.17	-6%
Hailfinder	56	0.02	0.02	0%
Medianus	56	0.04	0.03	25%
3Nt	58	0.02	0.01	50%
Hepar_Ii	70	0.03	0.03	0%
Win95Pts	76	0.03	0.03	0%
System_V57	85	0.06	0.05	17%
Fwe_Model8	109	0.14	0.15	-7%
Pathfinder	109	0.12	0.11	8%
Adapt_T1	133	0.04	0.04	0%
Cc145	145	0.1	0.08	20%
Munin1	189	0.54	0.75	-39%
Andes	223	0.15	0.13	13%
Cc245	245	0.2	0.18	10%
Diabetes	413	0.34	0.31	9%
Adapt_T2	671	0.24	0.22	8%
Amirali	681	0.45	0.41	9%
Munin2	1003	0.49	0.45	8%
Munin4	1041	0.61	0.57	7%
Munin3	1044	0.66	0.64	3%

- Experiments conducted on optimal JTs built from real-world and benchmark BNs
- SP was faster in 18/28
- SP tied LP in 5/28
- LP was faster in 5/28

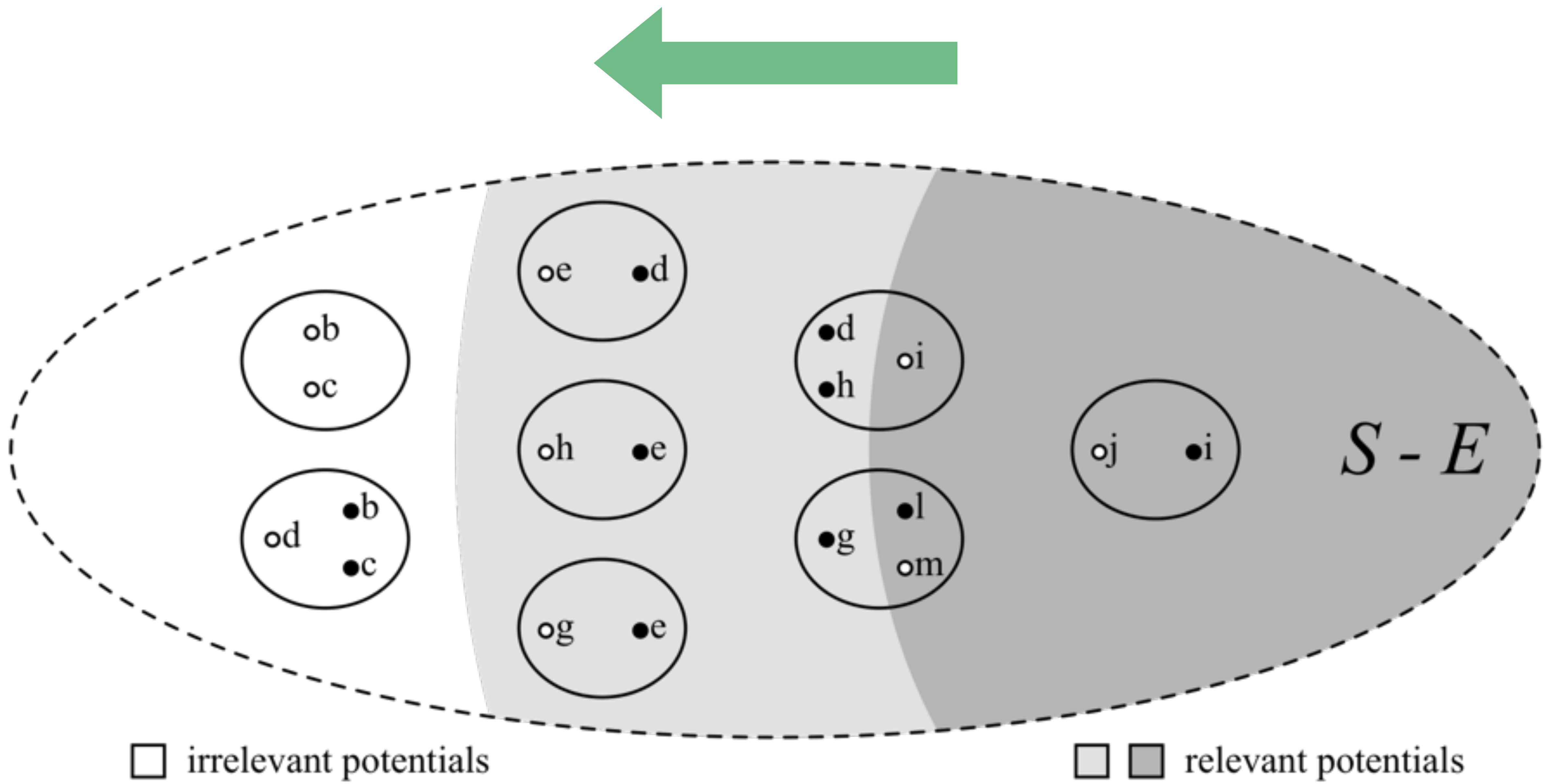
LP ANALYSIS

- Left-to-Right viewpoint



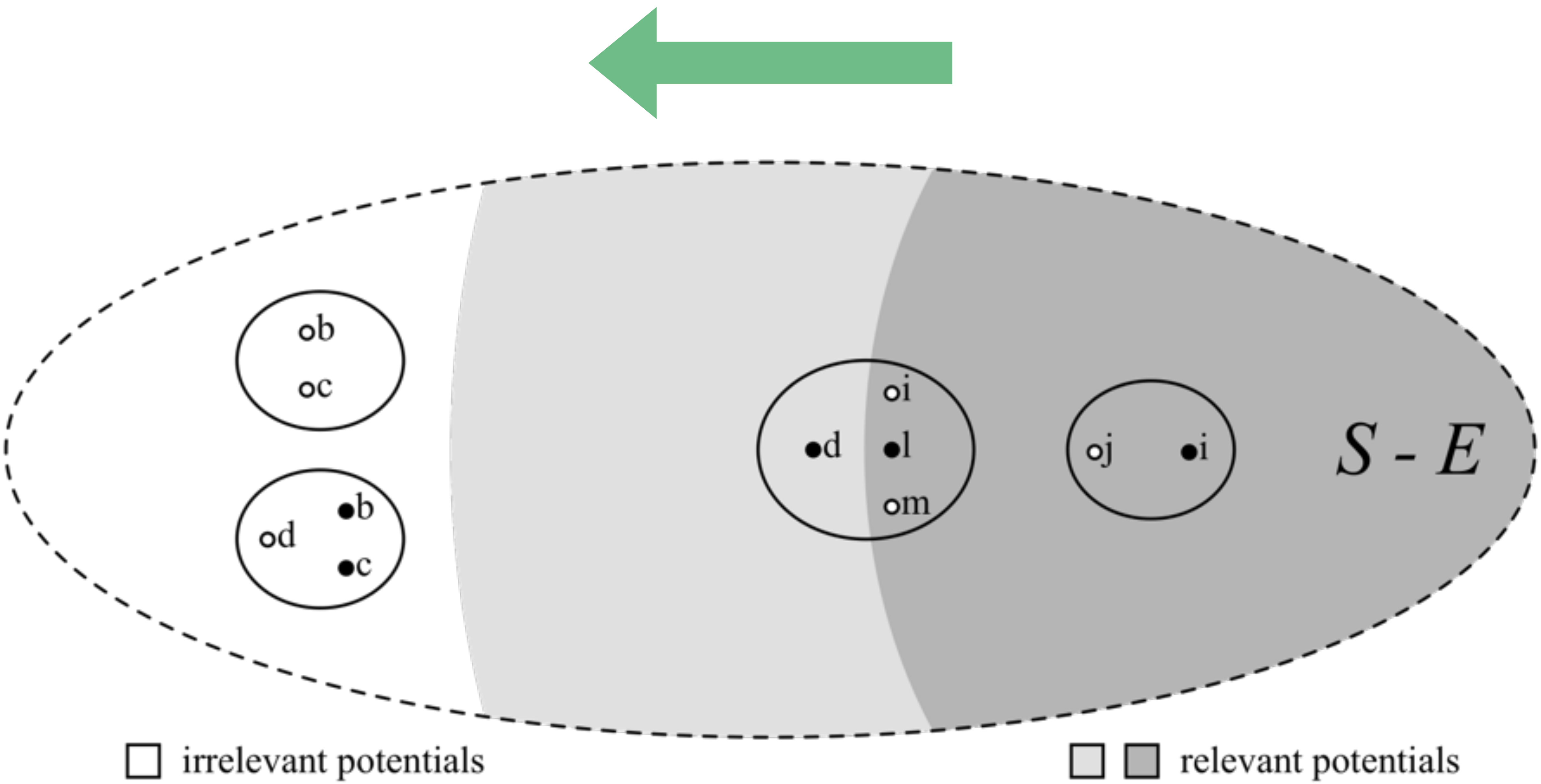
SP ANALYSIS

- Right-to-Left viewpoint



SP Analysis

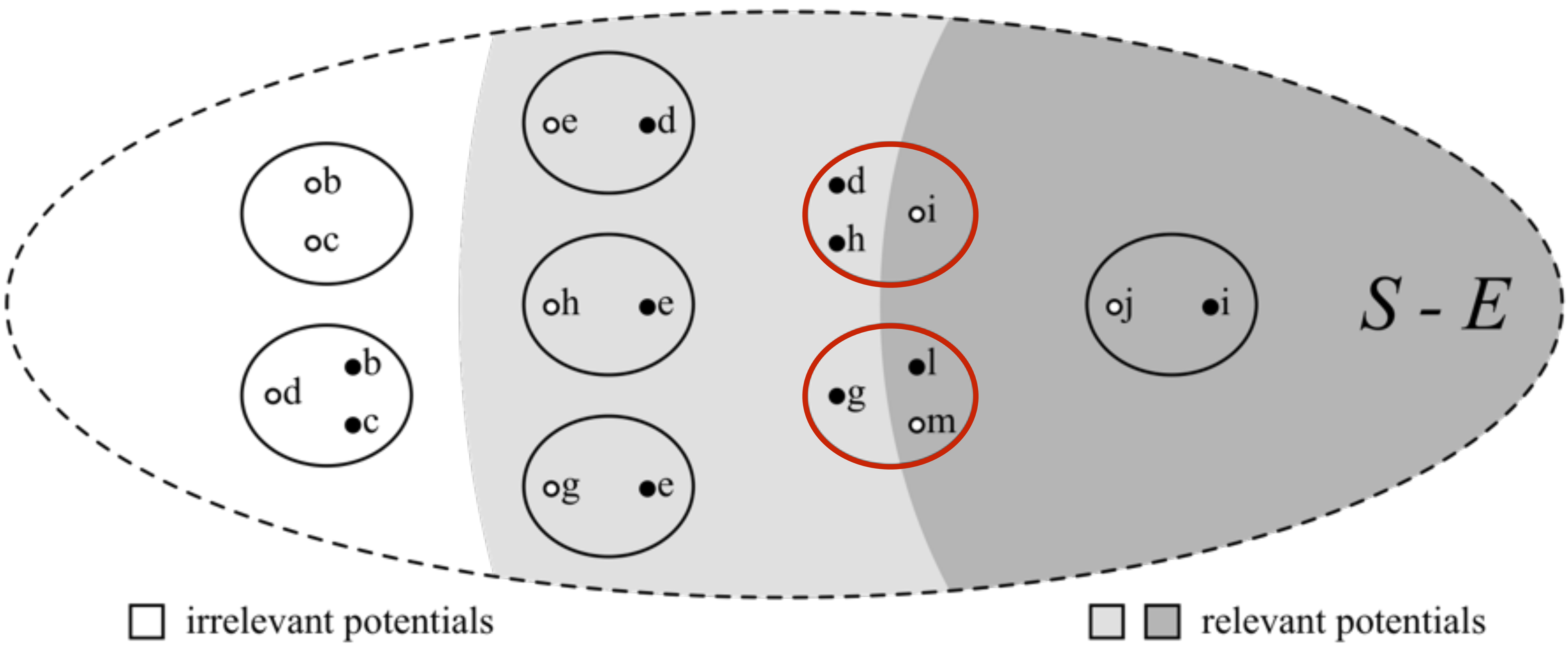
- Right-to-Left viewpoint



EXPERIMENTAL RESULTS

SP HEURISTICS

- SP is a new BN inference algorithm
- There may be more than one potential satisfying the **"one in, one out"** property



SP Heuristics

- Increasing variables in X (Inc X)
- Decreasing variables in X (Dec X)
- Increasing variables of X in S (Inc in S)
- Decreasing variables of X in S (Dec in S)
- Increasing variables in X size (Inc X Size)
- Decreasing variables in X size (Dec X Size)
- Increasing variables of X in S size (Inc in S Size)
- Decreasing variables of X in S size (Dec in S Size).

BN	Vars	Arbitrary Order	Inc X	Dec X	Inc in S	Dec in S	Inc X Size	Dec X Size	Inc in S Size	Dec in S Size
Water	32	0.05	0.04	0.05	0.04	0.05	0.04	0.05	0.04	0.05
oow	33	0.06	0.05	0.06	0.05	0.06	0.06	0.06	0.06	0.06
oow_bas	33	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Mildew	35	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
oow_solo	40	0.06	0.05	0.06	0.05	0.06	0.06	0.06	0.06	0.06
HKV	44	**0.02**	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Barley	48	0.09	0.09	0.09	0.09	0.09	0.09	0.08	0.09	0.08
KK	50	0.08	0.08	0.08	0.08	0.08	0.08	0.07	0.08	0.07
ship	50	0.16	0.13	0.16	0.13	0.16	0.15	0.15	0.15	0.15
hailfinder	56	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
medianus	56	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
3nt	58	**0.01**	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Hepar_II	70	0.24	0.23	0.32	0.28	0.31	0.23	0.32	0.27	0.31
win95pts	76	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
system_v57	85	0.05	0.05	0.06	0.05	0.06	0.05	0.05	0.05	0.05
FEW	109	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.14
pathfinder	109	0.09	0.09	0.09	0.09	0.09	0.10	0.09	0.09	0.10
Adapt_T1	133	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
cc145	145	0.08	0.08	0.08	0.07	0.08	0.08	0.08	0.08	0.07
Munin1	189	**0.68**	0.72	0.72	0.84	0.90	0.71	0.69	0.76	0.84
andes	223	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
cc245	245	0.17	0.18	0.18	0.17	0.18	0.18	0.17	0.18	0.17
Diabetes	413	**0.27**	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
Adapt_T2	671	0.19	0.19	0.19	0.20	0.20	0.19	0.20	0.20	0.20
Amirali	681	0.40	**0.39**	0.44	0.42	0.40	0.42	0.44	0.40	0.41
Munin2	1003	0.44	0.41	0.42	0.41	0.43	0.42	0.44	0.40	**0.39**
Munin4	1041	0.51	0.52	0.51	0.51	0.52	0.53	0.51	0.52	0.51
Munin3	1044	**0.53**	0.55	0.56	0.56	0.56	0.55	0.56	0.55	0.55
sacso	2371	**0.72**	0.75	0.74	0.74	0.74	0.75	0.74	0.74	0.75
Tied for first		13	16	11	17	9	12	14	11	13
Unique wins		**6**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**

ANALYSIS

- Our experimental results suggest that SP does not require elimination orderings, provided that an optimal (or close to) join tree is built from the real-world BNs

- It is possible that elimination orderings are needed for larger BNs or when non-optimal join trees are used, since SP's performance degrades dramatically when applied on non-optimal join trees (Madsen et al., 2016 Canadian AI)

CONCLUSION

- SP is a new BN inference algorithm
- **“one in, one out”** property
- SP is faster than LP in optimal join trees
- Our heuristics were slower than choosing potentials arbitrarily